W3School Html 参考手册

wizardforcel

Published with GitBook



目錄

W3School HTML 参考手册

来源：HTML 参考手册

整理：飞龙

HTML 标签列表

按字母顺序排列

New : HTML5 中的新标签。

标签	描述
<!--...-->	定义注释。
<!DOCTYPE>	定义文档类型。
<a>	定义锚。
<abbr>	定义缩写。
<acronym>	定义只取首字母的缩写。
<address>	定义文档作者或拥有者的联系信息。
<applet>	不赞成使用。定义嵌入的 applet。
<area>	定义图像映射内部的区域。
<article>	定义文章。
<aside>	定义页面内容之外的内容。
<audio>	定义声音内容。
<b>	定义粗体字。
<base>	定义页面中所有链接的默认地址或默认目标。
<basefont>	不赞成使用。定义页面中文本的默认字体、颜色或尺寸。
<bdi>	定义文本的文本方向，使其脱离其周围文本的方向设置。
<bdo>	定义文字方向。
<big>	定义大号文本。
<blockquote>	定义长的引用。
<body>	定义文档的主体。
 	定义简单的折行。
<button>	定义按钮 (push button)。
<canvas>	定义图形。
<caption>	定义表格标题。
<center>	不赞成使用。定义居中文本。

<cite>	定义引用(citation)。
<code>	定义计算机代码文本。
<col>	定义表格中一个或多个列的属性值。
<colgroup>	定义表格中供格式化的列组。
<command>	定义命令按钮。
<datalist>	定义下拉列表。
<dd>	定义定义列表中项目的描述。
<del>	定义被删除文本。
<details>	定义元素的细节。
<dir>	不赞成使用。定义目录列表。
<div>	定义文档中的节。
<dfn>	定义定义项目。
<dialog>	定义对话框或窗口。
<dl>	定义定义列表。
<dt>	定义定义列表中的项目。
<em>	定义强调文本。
<embed>	定义外部交互内容或插件。
<fieldset>	定义围绕表单中元素的边框。
<figcaption>	定义 figure 元素的标题。
<figure>	定义媒介内容的分组，以及它们的标题。
<font>	不赞成使用。定义文字的字体、尺寸和颜色。
<footer>	定义 section 或 page 的页脚。
<form>	定义供用户输入的 HTML 表单。
<frame>	定义框架集的窗口或框架。
<frameset>	定义框架集。
<h1> to <h6>	定义 HTML 标题。
<head>	定义关于文档的信息。
<header>	定义 section 或 page 的页眉。
<hr>	定义水平线。
<html>	定义 HTML 文档。
<i>	定义斜体字。
<iframe>	定义内联框架。

<img>	定义图像。
<input>	定义输入控件。
<ins>	定义被插入文本。
<isindex>	不赞成使用。定义与文档相关的可搜索索引。
<kbd>	定义键盘文本。
<keygen>	定义生成密钥。
<label>	定义 input 元素的标注。
<legend>	定义 fieldset 元素的标题。
	定义列表的项目。
<link>	定义文档与外部资源的关系。
<map>	定义图像映射。
<mark>	定义有记号的文本。
<menu>	定义命令的列表或菜单。
<menuitem>	定义用户可以从弹出菜单调用的命令/菜单项目。
<meta>	定义关于 HTML 文档的元信息。
<meter>	定义预定义范围内的度量。
<nav>	定义导航链接。
<noframes>	定义针对不支持框架的用户的替代内容。
<noscript>	定义针对不支持客户端脚本的用户的替代内容。
<object>	定义内嵌对象。
	定义有序列表。
<optgroup>	定义选择列表中相关选项的组合。
<option>	定义选择列表中的选项。
<output>	定义输出的一些类型。
<p>	定义段落。
<param>	定义对象的参数。
<pre>	定义预格式文本。
<progress>	定义任何类型的任务的进度。
<q>	定义短的引用。
<rp>	定义若浏览器不支持 ruby 元素显示的内容。
<rt>	定义 ruby 注释的解释。

<ruby>	定义 ruby 注释。
<s>	不赞成使用。定义加删除线的文本。
<samp>	定义计算机代码样本。
<script>	定义客户端脚本。
<section>	定义 section。
<select>	定义选择列表（下拉列表）。
<small>	定义小号文本。
<source>	定义媒介源。
<span>	定义文档中的节。
<strike>	不赞成使用。定义加删除线文本。
<strong>	定义强调文本。
<style>	定义文档的样式信息。
<sub>	定义下标文本。
<summary>	为 <details> 元素定义可见的标题。
<sup>	定义上标文本。
<table>	定义表格。
<tbody>	定义表格中的主体内容。
<td>	定义表格中的单元。
<textarea>	定义多行的文本输入控件。
<tfoot>	定义表格中的表注内容（脚注）。
<th>	定义表格中的表头单元格。
<thead>	定义表格中的表头内容。
<time>	定义日期/时间。
<title>	定义文档的标题。
<tr>	定义表格中的行。
<track>	定义用在媒体播放器中的文本轨道。
<tt>	定义打字机文本。
<u>	不赞成使用。定义下划线文本。
	定义无序列表。
<var>	定义文本的变量部分。
<video>	定义视频。
<wbr>	定义视频。

<xmp>	不赞成使用。定义预格式文本。

按功能类别排列

New : HTML5 中的新标签。

基础

标签	描述
<!DOCTYPE>	定义文档类型。
<html>	定义 HTML 文档。
<title>	定义文档的标题。
<body>	定义文档的主体。
<h1> to <h6>	定义 HTML 标题。
<p>	定义段落。
 	定义简单的折行。
<hr>	定义水平线。
<!--...-->	定义注释。

格式

标签	描述
<acronym>	定义只取首字母的缩写。
<abbr>	定义缩写。
<address>	定义文档作者或拥有者的联系信息。
<b>	定义粗体文本。
<bdi>	定义文本的文本方向，使其脱离其周围文本的方向设置。
<bdo>	定义文字方向。
<big>	定义大号文本。
<blockquote>	定义长的引用。
<center>	不赞成使用。定义居中文本。
<cite>	定义引用(citation)。
<code>	定义计算机代码文本。

<del>	定义被删除文本。
<dfn>	定义定义项目。
<em>	定义强调文本。
<font>	不赞成使用。定义文本的字体、尺寸和颜色
<i>	定义斜体文本。
<ins>	定义被插入文本。
<kbd>	定义键盘文本。
<mark>	定义有记号的文本。
<meter>	定义预定义范围内的度量。
<pre>	定义预格式文本。
<progress>	定义任何类型的任务的进度。
<q>	定义短的引用。
<rp>	定义若浏览器不支持 ruby 元素显示的内容。
<rt>	定义 ruby 注释的解释。
<ruby>	定义 ruby 注释。
<s>	不赞成使用。定义加删除线的文本。
<samp>	定义计算机代码样本。
<small>	定义小号文本。
<strike>	不赞成使用。定义加删除线文本。
<strong>	定义语气更为强烈的强调文本。
<sup>	定义上标文本。
<sub>	定义下标文本。
<time>	定义日期/时间。
<tt>	定义打字机文本。
<u>	不赞成使用。定义下划线文本。
<var>	定义文本的变量部分。
<wbr>	定义视频。

表单

标签	描述
<form>	定义供用户输入的 HTML 表单。
<input>	定义输入控件。
<textarea>	定义多行的文本输入控件。
<button>	定义按钮。
<select>	定义选择列表（下拉列表）。
<optgroup>	定义选择列表中相关选项的组合。
<option>	定义选择列表中的选项。
<label>	定义 input 元素的标注。
<fieldset>	定义围绕表单中元素的边框。
<legend>	定义 fieldset 元素的标题。
<isindex>	不赞成使用。定义与文档相关的可搜索索引。
<datalist>	定义下拉列表。
<keygen>	定义生成密钥。
<output>	定义输出的一些类型。

框架

标签	描述
<frame>	定义框架集的窗口或框架。
<frameset>	定义框架集。
<noframes>	定义针对不支持框架的用户的替代内容。
<iframe>	定义内联框架。

图像

标签	描述
<img>	定义图像。
<map>	定义图像映射。
<area>	定义图像地图内部的区域。
<canvas>	定义图形。
<figcaption>	定义 figure 元素的标题。
<figure>	定义媒介内容的分组，以及它们的标题。

音频/视频

标签	描述
<audio>	定义声音内容。
<source>	定义媒介源。
<track>	定义用在媒体播放器中的文本轨道。
<video>	定义视频。

链接

标签	描述
<a>	定义锚。
<link>	定义文档与外部资源的关系。
<nav>	定义导航链接。

列表

标签	描述
	定义无序列表。
	定义有序列表。
	定义列表的项目。
<dir>	不赞成使用。定义目录列表。
<dl>	定义定义列表。
<dt>	定义定义列表中的项目。
<dd>	定义定义列表中项目的描述。
<menu>	定义命令的菜单/列表。
<menuitem>	定义用户可以从弹出菜单调用的命令/菜单项目。
<command>	定义命令按钮。

表格

标签	描述
<table>	定义表格
<caption>	定义表格标题。
<th>	定义表格中的表头单元格。
<tr>	定义表格中的行。
<td>	定义表格中的单元。
<thead>	定义表格中的表头内容。
<tbody>	定义表格中的主体内容。
<tfoot>	定义表格中的表注内容（脚注）。
<col>	定义表格中一个或多个列的属性值。
<colgroup>	定义表格中供格式化的列组。

样式/节

标签	描述
<style>	定义文档的样式信息。
<div>	定义文档中的节。
<span>	定义文档中的节。
<header>	定义 section 或 page 的页眉。
<footer>	定义 section 或 page 的页脚。
<section>	定义 section。
<article>	定义文章。
<aside>	定义页面内容之外的内容。
<details>	定义元素的细节。
<dialog>	定义对话框或窗口。
<summary>	为 <details> 元素定义可见的标题。

元信息

标签	描述
<head>	定义关于文档的信息。
<meta>	定义关于 HTML 文档的元信息。
<base>	定义页面中所有链接的默认地址或默认目标。
<basefont>	不赞成使用。定义页面中文本的默认字体、颜色或尺寸。

编程

标签	描述
<script>	定义客户端脚本。
<noscript>	定义针对不支持客户端脚本的用户的替代内容。
<applet>	不赞成使用。定义嵌入的 applet。
<embed>	为外部应用程序（非 HTML）定义容器。
<object>	定义嵌入的对象。
<param>	定义对象的参数。

HTML <!--...--> 标签

实例

HTML 注释：

```
<!--这是一段注释。注释不会在浏览器中显示。-->

<p>这是一段普通的段落。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持注释标签。

定义和用法

注释标签用于在源代码中插入注释。注释不会显示在浏览器中。

您可使用注释对您的代码进行解释，这样做有助于您在以后的时间对代码的编辑。当您编写了大量代码时尤其有用。

使用注释标签来隐藏浏览器不支持的脚本也是一个好习惯（这样就不会把脚本显示为纯文本）：

```
<script type="text/javascript">
<!--
function displayMsg()
{
alert("Hello World!")
}
//-->
</script>
```

注释：注释行结尾处的两条斜杠 (//) 是 JavaScript 注释符号。这可以避免 JavaScript 执行 --> 标签。

HTML 4.01 与 HTML5 之间的差异

无。

标准属性

注释标签不支持任何标准属性。

如需更多有关 HTML 标准属性的信息，请访问 标准属性。

事件属性

注释标签不支持任何事件属性。

如需更多有关 HTML 事件属性的信息，请访问 事件属性。

HTML <!DOCTYPE> 标签

实例

```
<!DOCTYPE html>
<html>
<head>
<title>文档的标题</title>
</head>

<body>
文档的内容......
</body>

</html>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <!DOCTYPE> 声明。

定义和用法

<!DOCTYPE> 声明必须是 HTML 文档的第一行，位于 <html> 标签之前。

<!DOCTYPE> 声明不是 HTML 标签；它是指示 web 浏览器关于页面使用哪个 HTML 版本进行编写的指令。

在 HTML 4.01 中，<!DOCTYPE> 声明引用 DTD，因为 HTML 4.01 基于 SGML。DTD 规定了标记语言的规则，这样浏览器才能正确地呈现内容。

HTML5 不基于 SGML，所以不需要引用 DTD。

提示：请始终向 HTML 文档添加 <!DOCTYPE> 声明，这样浏览器才能获知文档类型。

HTML 4.01 与 HTML5 之间的差异

在 HTML 4.01 中有三种 <!DOCTYPE> 声明。在 HTML5 中只有一种：

```
<!DOCTYPE html>
```

HTML 元素和文档类型（Doctype）

请参阅这个 HTML 元素表，其中列出了每种元素会出现在哪个文档类型中。

提示和注释

注释：<!DOCTYPE> 声明没有结束标签。

提示：<!DOCTYPE> 声明对大小写不敏感。

提示：请使用 W3C 的验证器来检查您是否编写了有效的 HTML / XHTML 文档！

常用的 DOCTYPE 声明

HTML 5

```
<!DOCTYPE html>
```

HTML 4.01 Strict

该 DTD 包含所有 HTML 元素和属性，但不包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"
```

HTML 4.01 Transitional

该 DTD 包含所有 HTML 元素和属性，包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML 4.01 Frameset

该 DTD 等同于 HTML 4.01 Transitional，但允许框架集内容。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

XHTML 1.0 Strict

该 DTD 包含所有 HTML 元素和属性，但不包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）。必须以格式正确的 XML 来编写标记。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML 1.0 Transitional

该 DTD 包含所有 HTML 元素和属性，包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）。必须以格式正确的 XML 来编写标记。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML 1.0 Frameset

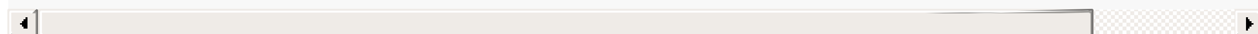
该 DTD 等同于 XHTML 1.0 Transitional，但允许框架集内容。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.1

该 DTD 等同于 XHTML 1.0 Strict，但允许添加模型（例如提供对东亚语系的 ruby 支持）。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml
```



HTML <a> 标签

实例

指向 w3school 的超链接：

```
<a href="http://www.w3school.com.cn">W3School</a>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <a> 标签。

定义和用法

<a> 标签定义超链接，用于从一张页面链接到另一张页面。

<a> 元素最重要的属性是 href 属性，它指示链接的目标。

在所有浏览器中，链接的默认外观是：

- 未被访问的链接带有下划线而且是蓝色的
- 已被访问的链接带有下划线而且是紫色的
- 活动链接带有下划线而且是红色的

提示：您可能已经注意到了，W3School 站点内的链接外观与默认的链接外观非常不同。您可以使用 CSS 伪类 向文本超链接添加复杂而多样的样式。

术语解释

什么是超文本？

提示和注释

提示：如果不使用 href 属性，则不可以使用如下属性：download, hreflang, media, rel, target 以及 type 属性。

提示：被链接页面通常显示在当前浏览器窗口中，除非您规定了另一个目标（target 属性）。

提示：请使用 CSS 来设置链接的样式。

HTML 4.01 与 HTML5 之间的差异

在 HTML 4.01 中，<a> 标签可以是超链接或锚。在 HTML5 中，<a> 标签始终是超链接，但是如果未设置 href 属性，则只是超链接的占位符。

HTML5 提供了一些新属性，同时不再支持一些 HTML 4.01 属性。

属性

New : HTML5 中的新属性。

属性	值	描述
charset	*char_encoding*	HTML5 中不支持。规定被链接文档的字符集。
coords	*coordinates*	HTML5 中不支持。规定链接的坐标。
download	*filename*	规定被下载的超链接目标。
href	*URL*	规定链接指向的页面的 URL。
hreflang	*language_code*	规定被链接文档的语言。
media	*media_query*	规定被链接文档是为何种媒介/设备优化的。
name	*section_name*	HTML5 中不支持。规定锚的名称。
rel	*text*	规定当前文档与被链接文档之间的关系。
rev	*text*	HTML5 中不支持。规定被链接文档与当前文档之间的关系。
shape	`default` `rect` `circle` `poly`	HTML5 中不支持。规定链接的形状。
target	`_blank` `_parent` `_self` `_top` `framename`	规定在何处打开链接文档。
type	*MIME type*	规定被链接文档的的 MIME 类型。

全局属性

<a> 标签支持 HTML 中的全局属性。

事件属性

<a> 标签支持 HTML 中的事件属性。

亲自试一试 - 实例

创建超级链接

本例演示如何在 HTML 文档中创建链接。

```
<html>

<body>

<p>
<a href="/index.html">本文本</a> 是一个指向本网站中的一个页面的链接。</p>

<p><a href="http://www.microsoft.com/">本文本</a> 是一个指向万维网上的页面的链接。</p>

</body>
</html>
```

将图像作为链接

本例演示如何使用图像作为链接。

```
<html>

<body>
<p>
您也可以使用图像来作链接：
<a href="/example/html/lastpage.html">
<img border="0" src="/i/eg_buttonnext.gif" />
</a>
</p>

</body>
</html>
```

链接到同一个页面的不同位置

本例演示如何使用链接跳转至文档的另一个部分

```
<html>

<body>

<p>
<a href="#C4">查看 Chapter 4。</a>
</p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2><a name="C4">Chapter 4</a></h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 10</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 14</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>
<p>This chapter explains ba bla bla</p>

</body>
</html>
```

在新的浏览器窗口打开链接

本例演示如何在新窗口打开一个页面，这样的话访问者就无需离开你的站点了。

```
<html>
```

```
<body>

<p>
<a href="#C4">查看 Chapter 4。</a>
</p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2><a name="C4">Chapter 4</a></h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 10</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 14</h2><html>

<body>

<p>
<a href="#C4">查看 Chapter 4。</a>
</p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2><a name="C4">Chapter 4</a></h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
<p>This chapter explains ba bla bla</p>
```

```
<h2>Chapter 7</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 10</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 14</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>
<p>This chapter explains ba bla bla</p>

</body>
</html>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>
<p>This chapter explains ba bla bla</p>

</body>
</html>
```

跳出框架

本例演示如何跳出框架，假如你的页面被固定在框架之内。

```
<html>

<body>

<p>被锁在框架中了吗？</p>

<a href="/index.html"
target="_top">请点击这里！</a>

</body>
</html>
```

创建电子邮件链接

本例演示如何如何链接到一个邮件。（本例在安装邮件客户端程序后才能工作。）

```
<html>

<body>

<p>
这是邮件链接：
<a href="mailto:someone@microsoft.com?subject=Hello%20again">发送邮件</a>
</p>

<p>
<b>注意：</b>应该使用 %20 来替换单词之间的空格，这样浏览器就可以正确地显示文本了。
</p>

</body>
</html>
```

创建电子邮件链接 2

本例演示更加复杂的邮件链接。

```
<html>

<body>

<p>
这是另一个 mailto 链接：
<a href="mailto:someone@microsoft.com?cc=someoneelse@microsoft.com&bcc=andsomeoneelse2@mi
</p>

<p>
<b>注意：</b>应该使用 %20 来替换单词之间的空格，这样浏览器就可以正确地显示文本了。
</p>

</body>
</html>
```

相关页面

HTML 教程：HTML 链接

HTML DOM 参考手册：Anchor 对象

HTML <abbr> 标签

实例

标记一个缩写：

```
The <abbr title="People's Republic of China">PRC</abbr> was founded in 1949.
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <abbr> 标签。

注释：IE 6 或更早版本的 IE 浏览器不支持 <abbr> 标签。

定义和用法

<abbr> 标签指示简称或缩写，比如 "WWW" 或 "NATO"。

通过对缩写进行标记，您能够为浏览器、拼写检查和搜索引擎提供有用的信息。

<abbr> 标签最初是在 HTML 4.0 中引入的，表示它所包含的文本是一个更长的单词或短语的缩写形式。

提示和注释

提示：可以在 <abbr> 标签中使用全局的 title 属性，这样就能够在鼠标指针移动到 <abbr> 元素上时显示出简称/缩写的完整版本。

HTML 4.01 与 HTML5 之间的差异

无。

全局属性

<abbr> 标签支持 HTML 中的全局属性。

事件属性

<abbr> 标签支持 HTML 中的事件属性。

HTML <acronym> 标签

实例

标记一个首字母缩写：

```
<acronym title="World Wide Web">WWW</acronym>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流的浏览器均支持 <acronym> 标签。

注释：IE 5.5 或更早版本的 IE 浏览器不支持 <acronym> 标签。

定义和用法

HTML5 中不支持 <acronym> 标签。请使用 <abbr> 标签代替。

<acronym> 标签定义首字母缩写。

缩写能够作为单词来朗读，例如 NATO, NASA, ASAP, GUI。

通过对缩写进行标记，您能够为浏览器、拼写检查和搜索引擎提供有用的信息。

提示和注释

提示：title 属性可用于在鼠标指针移动到元素上时显示出缩写的完整版本。

HTML 4.01 与 HTML5 之间的差异

HTML5 中不支持 <acronym> 标签，但是在 HTML 4.01 中支持。

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

如需完整的描述，请访问事件属性。

TIY 实例

缩写和首字母缩写

此例演示如何实现缩写或首字母缩写。

```
<html>

<body>

<abbr title="etcetera">etc.</abbr>
<br />
<acronym title="World Wide Web">WWW</acronym>

<p>在某些浏览器中，当您把鼠标移至缩略词语上时，title 可用于展示表达的完整版本。</p>

<p>仅对于 IE 5 中的 acronym 元素有效。</p>

<p>对于 Netscape 6.2 中的 abbr 和 acronym 元素都有效。</p>

</body>
</html>
```

HTML <address> 标签

实例

Example.com 的联系信息：

```
<address>
Written by <a href="mailto:webmaster@example.com">Donald Duck</a>.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <address> 标签。

定义和用法

<address> 标签定义文档或文章的作者/拥有者的联系信息。

如果 <address> 元素位于 <body> 元素内，则它表示文档联系信息。

如果 <address> 元素位于 <article> 元素内，则它表示文章的联系信息。

<address> 元素中的文本通常呈现为斜体。大多数浏览器会在 address 元素前后添加折行。

提示和注释

提示：<address> 标签不应该用于描述通讯地址，除非它是联系信息的一部分。

提示：<address> 元素通常连同其他信息被包含在 <footer> 元素中。

HTML 4.01 与 HTML5 之间的差异

HTML 4.01 不支持 <article> 标签，因此在 HTML 4.01 中，<address> 标签始终定义文档作者/拥有者的联系信息。

全局属性

<address> 标签支持 HTML 中的全局属性。

事件属性

<address> 标签支持 HTML 中的事件属性。

HTML <applet> 标签

实例

一个嵌入的 Java applet：

```
<applet code="Bubbles.class" width="350" height="350">
Java applet that draws animated bubbles.
</applet>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

某些浏览器中依然存在对 <applet> 标签的一些支持，但是需要额外的插件和安装过程才能起作用。

定义和用法

HTML5 中不支持 <applet> 标签。请使用 object 元素 标签代替。

HTML 4.01 中不赞成使用 <applet> 元素。

<applet> 标签定义嵌入的 applet。

HTML 4.01 与 HTML5 之间的差异

在 HTML5 中，不支持 applet 元素；在 HTML 4.01 中，不赞成使用 applet 元素。

必需的属性

属性	值	描述
code	*URL*	规定 Java applet 的文件名。
object	*name*	定义了包含该 applet 的一系列版本的资源名称。

可选的属性

属性	值	描述
align	`left` `right` `top` `bottom` `middle` `baseline` `texttop` `absmiddle` `absbottom`	定义 applet 相对于周围元素的对齐方式。
alt	*text*	规定 applet 的替换文本。
archive	*URL*	规定档案文件的位置。
codebase	*URL*	规定 code 属性中指定的 applet 的基准 URL。
height	*pixels*	定义 applet 的高度
hspace	*pixels*	定义围绕 applet 的水平间隔。
name	*unique_name*	规定 applet 的名称（用在脚本中的）。
vspace	*pixels*	定义围绕 applet 的垂直间隔。
width	*pixels*	定义 applet 的宽度

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

如需完整的描述，请访问事件属性。

HTML <area> 标签

实例

带有可点击区域的图像映射：

```
<img src="planets.jpg" border="0" usemap="#planetmap" alt="Planets" />

<map name="planetmap" id="planetmap">
  <area shape="circle" coords="180,139,14" href ="venus.html" alt="Venus" />
  <area shape="circle" coords="129,161,10" href ="mercur.html" alt="Mercury" />
  <area shape="rect" coords="0,0,110,260" href ="sun.html" alt="Sun" />
</map>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <area> 标签。

定义和用法

<area> 标签定义图像映射中的区域（注：图像映射指得是带有可点击区域的图像）。

area 元素总是嵌套在 <map> 标签中。

注释：<img> 标签中的 usemap 属性与 map 元素 name 属性相关联，创建图像与映射之间的联系。

HTML 与 XHTML 之间的差异

在 HTML 中，<area> 没有结束标签。

在 XHTML 中，<area> 必须正确地关闭。

提示和注释：

注释：<img> 中的 *usemap* 属性可引用 <map> 中的 *id* 或 *name* 属性（由浏览器决定），所以我们需要同时向 <map> 添加 id 和 name 两个属性。

必需的属性

属性	值	描述
alt	*text*	定义此区域的替换文本。

可选的属性

属性	值	描述
coords	坐标值	定义可点击区域（对鼠标敏感的区域）的坐标。
href	*URL*	定义此区域的目标 URL。
nohref	nohref	从图像映射排除某个区域。
shape	`default` `rect` `circ` `poly`	定义区域的形状。
target	`_blank` `_parent` `_self` `_top`	规定在何处打开 href 属性指定的目标 URL。

全局属性

<area> 标签支持 HTML 中的全局属性。

事件属性

<area> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Area 对象

HTML <article> 标签

实例

```
<article>
  <h1>Internet Explorer 9</h1>
  <p>Windows Internet Explorer 9（简称 IE9）于 2011 年 3 月 14 日发布.....</p>
</article>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9, Firefox, Opera, Chrome 以及 Safari 支持 <article> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <article> 标签。

定义和用法

<article> 标签规定独立的自包含内容。

一篇文章应有其自身的意义，应该有可能独立于站点的其余部分对其进行分发。

<article> 元素的潜在来源：

- 论坛帖子
- 报纸文章
- 博客条目
- 用户评论

HTML 4.01 与 HTML 5 之间的差异

<article> 标签是 HTML 5 中的新标签。

全局属性

<article> 标签支持 HTML 中的全局属性。

事件属性

<article> 标签支持 HTML 中的事件属性。

HTML <aside> 标签

实例

```
<p>Me and my family visited The Epcot center this summer.</p>
<aside>
<h4>Epcot Center</h4>
The Epcot Center is a theme park in Disney World, Florida.
</aside>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <aside> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <aside> 标签。

定义和用法

<aside> 标签定义其所处内容之外的内容。

aside 的内容应该与附近的内容相关。

HTML 4.01 与 HTML 5 之间的差异

<aside> 标签是 HTML 5 的新标签。

提示和注释：

提示：<aside> 的内容可用作文章的侧栏。

全局属性

<aside> 标签支持 HTML 中的全局属性。

事件属性

<aside> 标签支持 HTML 中的事件属性。

HTML <audio> 标签

实例

一段简单的 HTML 5 音频：

```
<audio src="someaudio.wav">
您的浏览器不支持 audio 标签。
</audio>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <audio> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <audio> 标签。

定义和用法

<audio> 标签定义声音，比如音乐或其他音频流。

HTML 4.01 与 HTML 5 之间的差异

<audio> 标签是 HTML 5 的新标签。

提示和注释

提示：可以在开始标签和结束标签之间放置文本内容，这样老的浏览器就可以显示出不支持该标签的信息。

属性

new : HTML5 中的新属性。

属性	值	描述
autoplay	autoplay	如果出现该属性，则音频在就绪后马上播放。
controls	controls	如果出现该属性，则向用户显示控件，比如播放按钮。
loop	loop	如果出现该属性，则每当音频结束时重新开始播放。
muted	muted	规定视频输出应该被静音。
preload	preload	如果出现该属性，则音频在页面加载时进行加载，并预备播放。如果使用 "autoplay"，则忽略该属性。
src	*url*	要播放的音频的 URL。

全局属性

<audio> 标签支持 HTML 中的全局属性。

事件属性

<audio> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Audio 对象

HTML <b> 标签

实例

```
<p>这是普通文本 - <b>这是粗体文本</b>。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <b> 标签。

定义和用法

<b> 标签规定粗体文本。

HTML 4.01 与 HTML5 之间的差异

无。

提示和注释

注释：根据 HTML5 规范，在没有其他合适标签更合适时，才应该把 <b> 标签作为最后的选项。HTML5 规范声明：应该使用 <h1> - <h6> 来表示标题，使用 <em> 标签来表示强调的文本，应该使用 <strong> 标签来表示重要文本，应该使用 <mark> 标签来表示标注的/突出显示的文本。

提示：您也能够使用 CSS "font-weight" 属性来设置粗体文本。

全局属性

<b> 标签支持 HTML 中的全局属性。

事件属性

<b> 标签支持 HTML 中的事件属性。

HTML <base> 标签

实例

```
<head>
<base href="http://www.w3school.com.cn/i/" />
<base target="_blank" />
</head>

<body>
<img src="eg_smile.gif" />
<a href="http://www.w3school.com.cn">W3School</a>
</body>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <base> 标签。

定义和用法

<base> 标签为页面上的所有链接规定默认地址或默认目标。

通常情况下，浏览器会从当前文档的 URL 中提取相应的元素来填写相对 URL 中的空白。

使用 <base> 标签可以改变这一点。浏览器随后将不再使用当前文档的 URL，而使用指定的基本 URL 来解析所有的相对 URL。这其中包括 <a>、<img>、<link>、<form> 标签中的 URL。

HTML 与 XHTML 之间的差异

在 HTML 中，<base> 标签没有结束标签；在 XHTML 中，<base> 标签必须被正确地关闭。

提示和注释：

注释：<base> 标签必须位于 head 元素内部。

必需的属性

属性	值	描述
href	*URL*	规定页面中所有相对链接的基准 URL。

可选的属性

属性	值	描述
target	`_blank` `_parent` `_self` `_top` `framename`	在何处打开页面中所有的链接。

标准的属性和事件

NONE

相关页面

HTML DOM 参考手册：Base 对象

HTML <basefont> 标签

实例

规定页面上的默认字体颜色和字号：

```
<head>
<basefont color="red" size="5" />
</head>

<body>
<h1>This is a header</h1>
<p>This is a paragraph</p>
</body>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

只有 Internet Explorer 支持 <basefont> 标签。应该避免使用该标签。

定义和用法

<basefont> 标签定义基准字体。该标签可以为文档中的所有文本定义默认字体颜色、字体大小和字体系列。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，不赞成使用 basefont 元素；在 XHTML 1.0 Strict DTD 中，不支持 basefont 元素。

可选的属性

属性	值	描述
color	*rgb(x,x,x) #xxxxxx colorname*	不赞成使用。请使用样式取代它。规定文档中的默认文本颜色。
face	*font_family*	不赞成使用。请使用样式取代它。规定文档中的默认字体。
size	*number*	不赞成使用。请使用样式取代它。规定文档中的默认字体大小。

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
NONE
```

HTML <bdi> 标签

实例

把用户名从周围的文本方向设置中隔离出来：

```
<ul>
<li>Username <bdi>Bill</bdi>:80 points</li>
<li>Username <bdi>Steve</bdi>: 78 points</li>
</ul>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前只有 Firefox 和 Chrome 支持 <bdi> 标签。

定义和用法

bdi 指的是 bidi 隔离。

<bdi> 标签允许您设置一段文本，使其脱离其父元素的文本方向设置。

在发布用户评论或其他您无法完全控制的内容时，该标签很有用。

HTML 4.01 与 HTML 5 之间的差异

<bdi> 标签是 HTML5 中的新标签。

属性

属性	值	描述
dir	`ltr` `rtl` `auto`	可选。规定 <bdi> 元素内的文本的文本方向。默认值：auto。

全局属性

<bdi> 标签支持 HTML 中的全局属性。

事件属性

<bdi> 标签支持 HTML 中的事件属性。

HTML <bdo> 标签

实例

```
<bdo dir="rtl">Here is some text</bdo>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <bdo> 标签。

定义和用法

bdo 元素可覆盖默认的文本方向。

HTML 与 XHTML 之间的差异

NONE

可选的属性

属性	值	描述
dir	`ltr` `rtl`	定义文字的方向

全局属性

<bdo> 标签支持 HTML 中的全局属性。

事件属性

<bdo> 标签支持 HTML 中的事件属性。

TIY 实例

文字方向

此例演示如何改变文字的方向。

HTML <big> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <big> 标签。

定义和用法

<big> 标签呈现大号字体效果。

使用 <big> 标签可以很容易地放大字体。这简直不能再简单了：浏览器显示包含在 <big> 标签和其相应的 </big> 标签之间的文字时，其字体比周围的文字要大一号。但是，如果文字已经是最大号字体，这个 <big> 标签将不起任何作用。

更妙的是，可以嵌套 <big> 标签来放大文本。每一个 <big> 标签都可以使字体大一号，直到上限 7 号文本，正如字体模型所定义的那样。

但是使用 <big> 标签的时候还是要小心，因为浏览器总是很宽大地试图去理解各种标签，对于那些不支持 <big> 标签的浏览器来说，它经常将其认为是粗体字标签。

参阅

请参阅 HTML 5 参考手册，了解有关 <big> 标签的最新信息：

HTML 5 <big> 标签

HTML <blockquote> 标签

实例

标记长的引用：

```
<blockquote>
Here is a long quotation here is a long quotation here is a long quotation
here is a long quotation here is a long quotation here is a long quotation
here is a long quotation here is a long quotation here is a long quotation.
</blockquote>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流的浏览器均支持 <blockquote> 标签。

注释：没有浏览器能够正确地显示 cite 属性。

定义和用法

<blockquote> 标签定义块引用。

<blockquote> 与 </blockquote> 之间的所有文本都会从常规文本中分离出来，经常会在左、右两边进行缩进（增加外边距），而且有时会使用斜体。也就是说，块引用拥有它们自己的空间。

HTML 与 XHTML 之间的差异

没有。

提示和注释：

提示：请使用 q 元素来标记短的引用。

注释：如需把页面作为 strict XHTML 进行验证，那么 <blockquote> 元素必须包含块级元素，比如这样：

```
<blockquote>
<p>here is a long quotation here is a long quotation</p>
</blockquote>
```

可选的属性

属性	值	描述
cite	*URL*	规定引用的来源。

全局属性

<blockquote> 标签支持 HTML 中的全局属性。

事件属性

<blockquote> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Blockquote 对象

HTML <body> 标签

实例

一个简单的 HTML 文档，带有最基本的必需的元素：

```
<html>

<head>
  <title>文档的标题</title>
</head>

<body>
  文档的内容... ...
</body>

</html>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <body> 标签。

定义和用法

body 元素定义文档的主体。

body 元素包含文档的所有内容（比如文本、超链接、图像、表格和列表等等。）

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，所有 body 元素的“呈现属性”均不被赞成使用。

在 XHTML 1.0 Strict DTD 中，所有 body 元素的“呈现属性”均不被支持。

可选的属性

属性	值	描述
alink	*rgb(x,x,x)* *#xxxxxx* *colorname*	不赞成使用。请使用样式取代它。规定文档中活动链接（active link）的的颜色。
background	*URL*	不赞成使用。请使用样式取代它。规定文档的背景图像。
bgcolor	*rgb(x,x,x)* *#xxxxxx* *colorname*	不赞成使用。请使用样式取代它。规定文档的背景颜色。
link	*rgb(x,x,x)* *#xxxxxx* *colorname*	不赞成使用。请使用样式取代它。规定文档中未访问链接的默认颜色。
text	*rgb(x,x,x)* *#xxxxxx* *colorname*	不赞成使用。请使用样式取代它。规定文档中所有文本的颜色。
vlink	*rgb(x,x,x)* *#xxxxxx* *colorname*	不赞成使用。请使用样式取代它。规定文档中已被访问链接的颜色。

全局属性

<body> 标签支持 HTML 中的全局属性。

事件属性

<body> 标签支持 HTML 中的事件属性。

TIY 实例

一个简单的 HTML 文件

这个例子是一个很简单的 HTML 文件，使用了尽量少的 HTML 标签。它向您演示了 body 元素中的内容是如何被浏览器显示的。

```
<html>

<head>
<title>我的第一个 HTML 页面</title>
</head>

<body>
<p>body 元素的内容会显示在浏览器中。</p>
<p>title 元素的内容会显示在浏览器的标题栏中。</p>
</body>

</html>
```

背景颜色

本例演示如何为 HTML 页面添加背景颜色。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<meta http-equiv="Content-Language" content="zh-cn" />
</head>

<body bgcolor="yellow">
<h2>请看: 改变了颜色的背景。</h2>
</body>

</html>
```

背景图片

本例演示如何添加背景图片到 HTML 页面。

```
<html>

<body background="/i/eg_background.jpg">

<h3>图像背景</h3>

<p>gif 和 jpg 文件均可用作 HTML 背景。</p>

<p>如果图像小于页面，图像会进行重复。</p>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Column 对象

HTML
 标签

TIY 实例

换行

本例演示如何在文档中插入换行符。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持
 标签。

定义和用法

 可插入一个简单的换行符。

 标签是空标签（意味着它没有结束标签，因此这是错误的：
</br>）。在 XHTML 中，把结束标签放在开始标签中，也就是
。

请注意，
 标签只是简单地开始新的一行，而当浏览器遇到 <p> 标签时，通常会在相邻的段落之间插入一些垂直的间距。

clear 属性

如果您希望文本流在内联表格或图像的下一行继续输出，请使用 clear 属性，该属性有三个可选的值：left、right 或者 all，每个值都代表一个边界或两边的边界。

更多有关
 标签的 clear 属性的信息。

HTML 与 XHTML 之间的差异

在 HTML 中，
 标签没有结束标签。

在 XHTML 中，
 标签必须被正确地关闭，比如这样：
。

提示和注释：

注释：请使用
 来输入空行，而不是分割段落。

全局属性

 标签支持 HTML 中的全局属性。

事件属性

 标签支持 HTML 中的事件属性。

HTML <button> 标签

实例

以下代码标记了一个按钮：

```
<button type="button">Click Me!</button>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <button> 标签。

重要事项：如果在 HTML 表单中使用 button 元素，不同的浏览器会提交不同的值。Internet Explorer 将提交 <button> 与 <button/> 之间的文本，而其他浏览器将提交 value 属性的内容。请在 HTML 表单中使用 input 元素来创建按钮。

定义和用法

<button> 标签定义一个按钮。

在 button 元素内部，您可以放置内容，比如文本或图像。这是该元素与使用 input 元素创建的按钮之间的不同之处。

<button> 控件 与 <input type="button"> 相比，提供了更为强大的功能和更丰富的内容。<button> 与 </button> 标签之间的所有内容都是按钮的内容，其中包括任何可接受的正文内容，比如文本或多媒体内容。例如，我们可以在按钮中包括一个图像和相关的文本，用它们在按钮中创建一个吸引人的标记图像。

唯一禁止使用的元素是图像映射，因为它对鼠标和键盘敏感的动作会干扰表单按钮的行为。

请始终为按钮规定 type 属性。Internet Explorer 的默认类型是 "button"，而其他浏览器中（包括 W3C 规范）的默认值是 "submit"。

HTML 4.01 与 HTML 5 之间的差异

HTML 5 中的新属性：autofocus, form, formaction, formenctype, formmethod, formnovalidate 以及 formtarget。

提示和注释

注释：如果在 HTML 表单中使用 button 元素，不同的浏览器会提交不同的按钮值。请使用 input 元素在 HTML 表单中创建按钮。

属性

new : HTML5 中的新属性。

属性	值	描述
autofocus	autofocus	规定当页面加载时按钮应当自动地获得焦点。
disabled	disabled	规定应该禁用该按钮。
form	*form_name*	规定按钮属于一个或多个表单。
formaction	*url*	覆盖 form 元素的 action 属性。注释：该属性与 type="submit" 配合使用。
formenctype	见注释	覆盖 form 元素的 enctype 属性。注释：该属性与 type="submit" 配合使用。
formmethod	`get` `post`	覆盖 form 元素的 method 属性。注释：该属性与 type="submit" 配合使用。
formnovalidate	formnovalidate	覆盖 form 元素的 novalidate 属性。注释：该属性与 type="submit" 配合使用。
formtarget	`_blank` `_self` `_parent` `_top` `framename`	覆盖 form 元素的 target 属性。注释：该属性与 type="submit" 配合使用。
name	*button_name*	规定按钮的名称。
type	`button` `reset` `submit`	规定按钮的类型。
value	*text*	规定按钮的初始值。可由脚本进行修改。

注释：formenctype 属性可能的值：

- application/x-www-form-urlencoded
- multipart/form-data
- text/plain

全局属性

<button> 标签支持 HTML 中的全局属性。

事件属性

<button> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Button 对象

HTML <canvas> 标签

实例

如何通过 canvas 元素来显示一个红色的矩形：

```
<canvas id="myCanvas"></canvas>

<script type="text/javascript">

var canvas=document.getElementById('myCanvas');
var ctx=canvas.getContext('2d');
ctx.fillStyle='#FF0000';
ctx.fillRect(0,0,80,100);

</script>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <canvas> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <canvas> 标签。

定义和用法

<canvas> 标签定义图形，比如图表和其他图像。

<canvas> 标签只是图形容器，您必须使用脚本来绘制图形。

HTML 4.01 与 HTML 5 之间的差异

<canvas> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
height	*pixels*	设置 canvas 的高度。
width	*pixels*	设置 canvas 的宽度。

全局属性

<canvas> 标签支持 HTML 中的全局属性。

事件属性

<canvas> 标签支持 HTML 中的事件属性。

<canvas> 的历史

这个 HTML 元素是为了客户端矢量图形而设计的。它自己没有行为，但却把一个绘图 API 展现给客户端 JavaScript 以使脚本能够把想绘制的东西都绘制到一块画布上。

<canvas> 标记由 Apple 在 Safari 1.3 Web 浏览器中引入。对 HTML 的这一根本扩展的原因在于，HTML 在 Safari 中的绘图能力也为 Mac OS X 桌面的 Dashboard 组件所使用，并且 Apple 希望有一种方式在 Dashboard 中支持脚本化的图形。

Firefox 1.5 和 Opera 9 都跟随了 Safari 的引领。这两个浏览器都支持 <canvas> 标记。

我们甚至可以在 IE 中使用 <canvas> 标记，并在 IE 的 VML 支持的基础上用开源的 JavaScript 代码（由 Google 发起）来构建兼容性的画布。 参见：http://excanvas.sourceforge.net/。

<canvas> 的标准化的努力由一个 Web 浏览器厂商的非正式协会在推进，目前 <canvas> 已经成为 HTML 5 草案中一个正式的标签。 参见：http://www.whatwg.org/specs/web-apps/current-work/

<canvas> 标记和 SVG 以及 VML 之间的差异

<canvas> 标记和 SVG 以及 VML 之间的一个重要的不同是，<canvas> 有一个基于 JavaScript 的绘图 API，而 SVG 和 VML 使用一个 XML 文档来描述绘图。

这两种方式在功能上是等同的，任何一种都可以用另一种来模拟。从表面上看，它们很不相同，可是，每一种都有强项和弱点。例如，SVG 绘图很容易编辑，只要从其描述中移除元素就行。

要从同一图形的一个 <canvas> 标记中移除元素，往往需要擦掉绘图重新绘制它。

如何使用 <canvas> 标记绘图

大多数 Canvas 绘图 API 都没有定义在 <canvas> 元素本身上，而是定义在通过画布的 getContext() 方法获得的一个“绘图环境”对象上。

Canvas API 也使用了路径的表示法。但是，路径由一系列的方法调用来定义，而不是描述为字母和数字的字符串，比如调用 beginPath() 和 arc() 方法。

一旦定义了路径，其他的方法，如 fill()，都是对此路径操作。绘图环境的各种属性，比如 fillStyle，说明了这些操作如何使用。

注释：Canvas API 非常紧凑的一个原因上它没有对绘制文本提供任何支持。要把文本加入到一个 <canvas> 图形，必须要么自己绘制它再用位图图像合并它，或者在 <canvas> 上方使用 CSS 定位来覆盖 HTML 文本。

相关页面

HTML 教程：HTML <canvas> 元素

HTML DOM 参考手册：HTML DOM Canvas 对象

HTML <caption> 标签

实例

```
<table border="1">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <caption> 标签。

定义和用法

caption 元素定义表格标题。

caption 标签必须紧随 table 标签之后。您只能对每个表格定义一个标题。通常这个标题会被居中于表格之上。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，caption 元素的 align 属性是不被赞成使用的。

在 XHTML 1.0 Strict DTD 中，caption 元素的 align 属性是不被支持的。

可选的属性

属性	值	描述
align	`left` `right` `top` `bottom`	不赞成使用。请使用样式取而代之。规定标题的对齐方式。

全局属性

<caption> 标签支持 HTML 中的全局属性。

事件属性

<caption> 标签支持 HTML 中的事件属性。

HTML <center> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <center> 标签。

定义和用法

对其所包括的文本进行水平居中。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，center 元素不被赞成使用。

在 XHTML 1.0 Strict DTD 中，center 元素不被支持。

提示和注释：

注释：提示：请使用 CSS 样式来居中文本！

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

如需完整的描述，请访问事件属性。

HTML <cite> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <cite> 标签。

定义和用法

<cite> 标签通常表示它所包含的文本对某个参考文献的引用，比如书籍或者杂志的标题。

按照惯例，引用的文本将以斜体显示。

用 <cite> 标签把指向其他文档的引用分离出来，尤其是分离那些传统媒体中的文档，如书籍、杂志、期刊，等等。如果引用的这些文档有联机版本，还应该把引用包括在一个 <a> 标签中，从而把一个超链接指向该联机版本。

<cite> 标签还有一个隐藏的功能：它可以使你或者其他人从文档中自动摘录参考书目。我们可以很容易地想象一个浏览器，它能够自动整理引用表格，并把它们作为脚注或者独立的文档来显示。<cite> 标签的语义已经远远超过了改变它所包含的文本外观的作用；它使浏览器能够以各种实用的方式来向用户表达文档的内容。

HTML <em> <strong> <dfn> <code> <samp> <kbd><var> <cite> 标签

定义和用法

以下元素都是短语元素。虽然这些标签定义的文本大多会呈现出特殊的样式，但实际上，这些标签都拥有确切的语义。

我们并不反对使用它们，但是如果您只是为了达到某种视觉效果而使用这些标签的话，我们建议您使用样式表，那么做会达到更加丰富的效果。

| <em> | 把文本定义为强调的内容。 || <strong> | 把文本定义为语气更强的强调的内容。 || <dfn> | 定义一个定义项目。 || <code> | 定义计算机代码文本。 || <samp> | 定义样本文本。 || <kbd> | 定义键盘文本。它表示文本是从键盘上键入的。它经常用在与计算机相关的文档或手册中。 || <var> | 定义变量。您可以将此标签与 <pre> 及 <code> 标签配合使用。 || <cite> | 定义引用。可使用该标签对参考文献的引用进行定义，比如书籍或杂志的标题。 |

HTML 与 XHTML 之间的差异

无

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

如需完整的描述，请访问事件属性。

HTML 文本格式化实例

文本格式化

此例演示如何在一个 HTML 文件中对文本进行格式化

```
<html>

<body>

<b>This text is bold</b>

<br />

<strong>This text is strong</strong>

<br />

<big>This text is big</big>

<br />

<em>This text is emphasized</em>

<br />

<i>This text is italic</i>

<br />

<small>This text is small</small>

<br />

This text contains
<sub>subscript</sub>

<br />

This text contains
<sup>superscript</sup>

</body>
</html>
```

“计算机输出”标签

此例演示不同的“计算机输出”标签的显示效果。

```
<html>

<body>

<code>Computer code</code>
<br />
<kbd>Keyboard input</kbd>
<br />
<tt>Teletype text</tt>
<br />
<samp>Sample text</samp>
<br />
<var>Computer variable</var>
<br />

<p>
<b>注释：</b>这些标签常用于显示计算机/编程代码。
</p>

</body>
</html>
```

HTML <col> 标签

实例

col 元素为表格中的三个列规定了不同的对齐方式：

```
<table width="100%" border="1">
  <col align="left" />
  <col align="left" />
  <col align="right" />
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <col> 标签。

定义和用法

<col> 标签为表格中一个或多个列定义属性值。

如需对全部列应用样式，<col> 标签很有用，这样就不需要对各个单元和各行重复应用样式了。

您只能在 table 或 colgroup 元素中使用 <col> 标签。

HTML 与 XHTML 之间的差异

在 HTML 中，<col> 没有结束标签。

在 XHTML 中，<col> 标签必须被正确的关闭。

提示和注释：

提示：请为 <col> 标签添加 class 属性。这样就可以使用 CSS 来负责对齐方式、宽度和颜色等等。

提示：如果您希望在 colgroup 内部为每个列规定不同的属性值时，请使用此元素。如果没有 col 元素，列会从 colgroup 那里继承所有的属性值。

注释：col 元素是仅包含属性的空元素。如需创建列，您就必须在 tr 元素内部规定 td 元素。

参见：<colgroup> 标签。

可选的属性

属性	值	描述
align	`right` `left` `center` `justify` `char`	规定与 col 元素相关的内容的水平对齐方式。
char	*character*	规定根据哪个字符来对齐与 col 元素相关的内容。
charoff	*number*	规定第一个对齐字符的偏移量。
span	*number*	规定 col 元素应该横跨的列数。
valign	`top` `middle` `bottom` `baseline`	定义与 col 元素相关的内容的垂直对齐方式。
width	*pixels % relative_length*	规定 col 元素的宽度。

全局属性

<col> 标签支持 HTML 中的全局属性。

事件属性

<col> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Column 对象

HTML <colgroup> 标签

实例

两个 colgroup 元素为表格中的三列规定了不同的对齐方式和样式（注意第一个 colgroup 元素横跨两列）：

```
<table width="100%" border="1">
  <colgroup span="2" align="left"></colgroup>
  <colgroup align="right" style="color:#0000FF;"></colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <colgroup> 标签。

Firefox、Chrome 以及 Safari 仅支持 colgroup 元素的 span 和 width 属性。

定义和用法

<colgroup> 标签用于对表格中的列进行组合，以便对其进行格式化。

如需对全部列应用样式，<colgroup> 标签很有用，这样就不需要对各个单元和各行重复应用样式了。

<colgroup> 标签只能在 table 元素中使用。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

提示：请为 <colgroup> 标签添加 class 属性。这样就可以使用 CSS 来负责对齐方式、宽度和颜色等等。

注释：col 元素是仅包含属性的空元素。如需创建列，您就必须在 tr 元素内部规定 td 元素。

参见：<col> 标签。

可选的属性

属性	值	描述
align	right left center justify char	定义在列组合中内容的水平对齐方式。
char	*character*	规定根据哪个字符来对齐列组中的内容。
charoff	*number*	规定第一个对齐字符的偏移量。
span	*number*	规定列组应该横跨的列数。
valign	top middle bottom baseline	定义在列组合中内容的垂直对齐方式。
width	*pixels % relative_length*	规定列组合的宽度。

全局属性

<colgroup> 标签支持 HTML 中的全局属性。

事件属性

<colgroup> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Column 对象

HTML <command> 标签

实例

标记一个按钮：

```
<menu>
<command onclick="alert('Hello World')">
Click Me!</command>
</menu>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

没有浏览器支持 <command> 标签。

只有 Internet Explorer 9 （更早或更晚的版本都不支持）支持 <command> 标签。

定义和用法

command 元素表示用户能够调用的命令。

<command> 标签可以定义命令按钮，比如单选按钮、复选框或按钮。

只有当 command 元素位于 menu 元素内时，该元素才是可见的。否则不会显示这个元素，但是可以用它规定键盘快捷键。

浏览器支持

目前只有 Internet Explorer 支持 <command> 标签。

HTML 4.01 与 HTML 5 之间的差异

<command> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
checked	checked	定义是否被选中。仅用于 radio 或 checkbox 类型。
disabled	disabled	定义 command 是否可用。
icon	*url*	定义作为 command 来显示的图像的 url。
label	*text*	为 command 定义可见的 label。
radiogroup	*groupname*	定义 command 所属的组名。仅在类型为 radio 时使用。
type	`checkbox` `command` `radio`	定义该 command 的类型。默认是 "command"。

全局属性

<command> 标签支持 HTML 中的全局属性。

事件属性

<command> 标签支持 HTML 中的事件属性。

HTML <datalist> 标签

实例

下面是一个 input 元素，datalist 中描述了其可能的值：

```
<input id="myCar" list="cars" />
<datalist id="cars">
  <option value="BMW">
  <option value="Ford">
  <option value="Volvo">
</datalist>
```

定义和用法

<datalist> 标签定义选项列表。请与 input 元素配合使用该元素，来定义 input 可能的值。

datalist 及其选项不会被显示出来，它仅仅是合法的输入值列表。

请使用 input 元素的 list 属性来绑定 datalist。

浏览器支持

所有主流浏览器都支持 <datalist> 标签，除了 Internet Explorer 和 Safari。

HTML 4.01 与 HTML 5 之间的差异

<datalist> 标签是 HTML 5 中的新标签。

全局属性

<datalist> 标签支持 HTML 中的全局属性。

事件属性

<datalist> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Datalist 对象

HTML <dd> 标签

实例

```
<dl>
   <dt>计算机</dt>
   <dd>用来计算的仪器 ... ...</dd>
   <dt>显示器</dt>
   <dd>以视觉方式显示信息的装置 ... ...</dd>
</dl>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <dd> 标签。

定义和用法

<dd> 在定义列表中定义条目的定义部分。

HTML 与 XHTML 之间的差异

NONE

全局属性

<dd> 标签支持 HTML 中的全局属性。

事件属性

<dd> 标签支持 HTML 中的事件属性。

TIY 实例

定义列表

本例演示一个定义列表。

```
<html>

<body>

<h2>一个定义列表：</h2>

<dl>
   <dt>计算机</dt>
   <dd>用来计算的仪器 ... ...</dd>
   <dt>显示器</dt>
   <dd>以视觉方式显示信息的装置 ... ...</dd>
</dl>

</body>
</html>
```

HTML <del> 标签

实例

```
a dozen is <del>20</del> 12 pieces
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <del> 标签。

定义和用法

定义文档中已被删除的文本。

提示和注释：

注释：请与 <ins> 标签配合使用，来描述文档中的更新和修正。

HTML 与 XHTML 之间的差异

NONE

可选的属性

属性	值	描述
cite	*URL*	指向另外一个文档的 URL，此文档可解释文本被删除的原因。
datetime	*YYYYMMDD*	定义文本被删除的日期和时间。

全局属性

<del> 标签支持 HTML 中的全局属性。

事件属性

<del> 标签支持 HTML 中的事件属性。

TIY 实例

删除和插入文本

此例演示如何标记文档中删除和插入的文本。

```
<html>

<body>

<p>一打有 <del>二十</del> <ins>十二</ins> 件。</p>

<p>大多数浏览器会改写为删除文本和下划线文本。</p>

<p>一些老式的浏览器会把删除文本和下划线文本显示为普通文本。</p>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Del 对象

HTML <details> 标签

实例

关于文档的细节：

```
<details>
<summary>Copyright 2011.</summary>
<p>All pages and graphics on this web site are the property of W3School.</p>
</details>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前只有 Chrome 和 Safari 6 支持 <details> 标签。

定义和用法

<details> 标签用于描述文档或文档某个部分的细节。

浏览器支持

目前只有 Chrome 支持 <details> 标签。

HTML 4.01 与 HTML 5 之间的差异

<details> 标签是 HTML 5 中的新标签。

提示和注释

提示：与 <summary> 标签 配合使用可以为 details 定义标题。标题是可见的，用户点击标题时，会显示出 details。

属性

new : HTML5 中的新属性。

属性	值	描述
open	open	定义 details 是否可见。

全局属性

<details> 标签支持 HTML 中的全局属性。

事件属性

<details> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Details 对象

HTML <dialog> 标签

实例

使用 <dialog> 元素：

```
<table border="1">
<tr>
  <th>一月 <dialog open>这是打开的对话窗口</dialog></th>
  <th>二月</th>
  <th>三月</th>
</tr>
<tr>
  <td>31</td>
  <td>28</td>
  <td>31</td>
</tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前只有 Chrome 和 Safari 6 支持 <dialog> 标签。

定义和用法

<dialog> 标签定义对话框或窗口。

HTML 4.01 与 HTML 5 之间的差异

<dialog> 标签是 HTML 5 的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
open	open	规定 dialog 元素是活动的，用户可与之交互。

全局属性

<dialog> 标签支持 HTML 中的全局属性。

事件属性

<dialog> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Dialog 对象

HTML <dir> 标签

不赞成使用 **dir** 元素！

实例

目录列表：

```
<dir>
   <li>HTML</li>
   <li>XHTML</li>
   <li>CSS</li>
</dir>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <dir> 标签。

定义和用法

<dir> 标签定义目录列表。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，dir 元素不被赞成使用。

在 XHTML 1.0 Strict DTD 中，dir 元素不被支持。

提示和注释：

提示：请使用 CSS 来为列表添加样式！在我们的 CSS 教程中，您可以找到更多有关为列表添加样式的细节。

可选的属性

属性	值	描述
compact	compact	不赞成使用。请使用样式代替。

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
onclick, ondblclick, onfocus, onblur, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

如需完整的描述，请访问事件属性。

HTML <div> 标签

实例

文档中的一个部分会显示为绿色：

```
<div style="color:#00FF00">
  <h3>This is a header</h3>
  <p>This is a paragraph.</p>
</div>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <div> 标签。

定义和用法

<div> 可定义文档中的分区或节（division/section）。

<div> 标签可以把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。

如果用 id 或 class 来标记 <div>，那么该标签的作用会变得更加有效。

用法

<div> 是一个块级元素。这意味着它的内容自动地开始一个新行。实际上，换行是 <div> 固有的唯一格式表现。可以通过 <div> 的 class 或 id 应用额外的样式。

不必为每一个 <div> 都加上类或 id，虽然这样做也有一定的好处。

可以对同一个 <div> 元素应用 class 或 id 属性，但是更常见的情况是只应用其中一种。这两者的主要差异是，class 用于元素组（类似的元素，或者可以理解为某一类元素），而 id 用于标识单独的唯一的元素。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，div 元素的 "align" 属性不被赞成使用。

在 XHTML 1.0 Strict DTD 中，div 元素的 "align" 属性不被支持。

提示和注释：

注释：<div> 是一个块级元素，也就是说，浏览器通常会在 div 元素前后放置一个换行符。

提示：请使用 <div> 元素来组合块级元素，这样就可以使用样式对它们进行格式化。

案例分析

```
<body>

 <h1>NEWS WEBSITE</h1>
  <p>some text. some text. some text...</p>
  ...

 <div class="news">
  <h2>News headline 1</h2>
  <p>some text. some text. some text...</p>
  ...
 </div>

 <div class="news">
  <h2>News headline 2</h2>
  <p>some text. some text. some text...</p>
  ...
 </div>

 ...
</body>
```

例子解释

正如您看到的，上面这段 HTML 模拟了新闻网站的结构。其中的每个 div 把每条新闻的标题和摘要组合在一起，也就是说，div 为文档添加了额外的结构。同时，由于这些 div 属于同一类元素，所以可以使用 class="news" 对这些 div 进行标识，这么做不仅为 div 添加了合适的语义，而且便于进一步使用样式对 div 进行格式化，可谓一举两得。

提示：如需更深入地学习 class 和 id 的用法，请阅《W3school 的结构化标记》这一章中 div、id 和其他帮手 这一节。

可选的属性

属性	值	描述
align	`left` `right` `center` `justify`	不赞成使用。请使用样式取而代之。规定 div 元素中的内容的对齐方式。

全局属性

<div> 标签支持 HTML 中的全局属性。

事件属性

<div> 标签支持 HTML 中的事件属性。

HTML <dl> 标签

实例

```
<dl>
   <dt>计算机</dt>
   <dd>用来计算的仪器 ... ...</dd>
   <dt>显示器</dt>
   <dd>以视觉方式显示信息的装置 ... ...</dd>
</dl>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <dl> 标签。

定义和用法

<dl> 标签定义了定义列表（definition list）。

<dl> 标签用于结合 <dt> （定义列表中的项目）和 <dd> （描述列表中的项目）。

HTML 与 XHTML 之间的差异

NONE

全局属性

<dl> 标签支持 HTML 中的全局属性。

事件属性

<dl> 标签支持 HTML 中的事件属性。

HTML <dt> 标签

实例

```
<dl>
   <dt>计算机</dt>
   <dd>用来计算的仪器 ... ...</dd>
   <dt>显示器</dt>
   <dd>以视觉方式显示信息的装置 ... ...</dd>
</dl>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <dt> 标签。

定义和用法

<dt> 标签定义了定义列表中的项目（即术语部分）。

HTML 与 XHTML 之间的差异

NONE

全局属性

<dt> 标签支持 HTML 中的全局属性。

事件属性

<dt> 标签支持 HTML 中的事件属性。

TIY 实例

定义列表

如何创建定义列表。

HTML <embed> 标签

定义和用法

<embed> 标签定义嵌入的内容，比如插件。

实例

```
<embed src="helloworld.swf" />
```

HTML 4.01 与 HTML 5 之间的差异

<embed> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
height	*pixels*	设置嵌入内容的高度。
src	*url*	嵌入内容的 URL。
type	*type*	定义嵌入内容的类型。
width	*pixels*	设置嵌入内容的宽度。

全局属性

<embed> 标签支持 HTML 中的全局属性。

事件属性

<embed> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Embed 对象

HTML <fieldset> 标签

实例

组合表单中的相关元素：

```
<form>
  <fieldset>
    <legend>health information</legend>
    height: <input type="text" />
    weight: <input type="text" />
  </fieldset>
</form>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <fieldset> 标签。

定义和用法

fieldset 元素可将表单内的相关元素分组。

<fieldset> 标签将表单内容的一部分打包，生成一组相关表单的字段。

当一组表单元素放到 <fieldset> 标签内时，浏览器会以特殊方式来显示它们，它们可能有特殊的边界、3D 效果，或者甚至可创建一个子表单来处理这些元素。

<fieldset> 标签没有必需的或唯一的属性。

<legend> 标签为 fieldset 元素定义标题。

HTML 4.01 与 HTML 5 之间的差异

HTML5 拥有如下属性：disabled、form、name，HTML 4.01 中不支持这些属性。

属性

属性	值	描述
disabled	disabled	规定应该禁用 fieldset。
form	*form_id*	规定 fieldset 所属的一个或多个表单。
name	*value*	规定 fieldset 的名称。

全局属性

<fieldset> 标签支持 HTML 中的全局属性。

事件属性

<fieldset> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Fieldset 对象

TIY 实例

围绕数据的 Fieldset

本例演示如何在数据周围绘制一个带标题的框。

```
<!DOCTYPE HTML>
<html>

<body>

<form>
  <fieldset>
    <legend>健康信息</legend>
    身高：<input type="text" />
    体重：<input type="text" />
  </fieldset>
</form>

<p>如果表单周围没有边框，说明您的浏览器太老了。</p>

</body>
</html>
```

HTML <figcaption> 标签

实例

用作文档中插图的图像，带有一个标题：：

```
<figure>
  <figcaption>黄浦江上的的卢浦大桥</figcaption>
  <img src="shanghai_lupu_bridge.jpg" width="350" height="234" />
</figure>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9, Firefox, Opera, Chrome 以及 Safari 支持 <figcaption> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <figcaption> 标签。

定义和用法

<figcaption> 标签定义 figure 元素的标题（caption）。

"figcaption" 元素应该被置于 "figure" 元素的第一个或最后一个子元素的位置。

浏览器支持

所有主流浏览器都支持 <figcaption> 标签。

HTML 4.01 与 HTML 5 之间的差异

<figcaption> 标签是 HTML 5 中的新标签。

全局属性

<figcaption> 标签支持 HTML 中的全局属性。

事件属性

<figcaption> 标签支持 HTML 中的事件属性。

HTML <figure> 标签

实例

用作文档中插图的图像：

```
<figure>
  <p>黄浦江上的的卢浦大桥</p>
  <img src="shanghai_lupu_bridge.jpg" width="350" height="234" />
</figure>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9, Firefox, Opera, Chrome 以及 Safari 支持 <figure> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <figure> 标签。

定义和用法

<figure> 标签规定独立的流内容（图像、图表、照片、代码等等）。

figure 元素的内容应该与主内容相关，但如果被删除，则不应对文档流产生影响。

浏览器支持

所有主流浏览器都支持 <figure> 标签。

HTML 4.01 与 HTML 5 之间的差异

<figure> 标签是 HTML 5 中的新标签。

提示和注释

提示：请使用 <figcaption> 元素为 figure 添加标题（caption）。

全局属性

<figure> 标签支持 HTML 中的全局属性。

事件属性

<figure> 标签支持 HTML 中的事件属性。

HTML <font> 标签

定义和用法

<font> 规定文本的字体、字体尺寸、字体颜色。

实例

规定文本字体、大小和颜色：

```
<font size="3" color="red">This is some text!</font>
<font size="2" color="blue">This is some text!</font>
<font face="verdana" color="green">This is some text!</font>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <font> 标签。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，font 元素不被赞成使用。

在 XHTML 1.0 Strict DTD 中，font 元素不被支持。

提示和注释：

提示：请使用样式（代替 <font>）来定义文本的字体、字体颜色、字体尺寸。

可选的属性

属性	值	描述
color	*rgb(x,x,x) #xxxxxx colorname*	不赞成使用。请使用样式取代它。规定文本的颜色。
face	*font_family*	不赞成使用。请使用样式取代它。规定文本的字体。
size	*number*	不赞成使用。请使用样式取代它。规定文本的大小。

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
NONE
```

如需完整的描述，请访问事件属性。

TIY 实例

设置文字的字体

本例演示如何设置文字的字体。

```
<html>
<body>
<h1 style="font-family:verdana">A heading</h1>
<p style="font-family:courier">A paragraph</p>
</body>
</html>
```

设置文字的尺寸

本例演示如何设置文字的尺寸。

```
<html>
<body>
<h1 style="font-size:150%">A heading</h1>
<p style="font-size:80%">A paragraph</p>
</body>
</html>
```

设置文字的颜色

本例演示如何设置文字的颜色。

```
<html>
<body>
<h1 style="color:blue">A heading</h1>
<p style="color:red">A paragraph</p>
</body>
</html>
```

设置文字的字体、字体尺寸、字体颜色

本例演示如何设置文字的字体、字体尺寸、字体颜色.

```
<html>
<body>
<p style="font-family:verdana;font-size:80%;color:green">
This is a paragraph with some text in it. This is a paragraph with some text in it. This
</p>
</body>
</html>
```



HTML <footer> 标签

实例

文档中的页脚部分：

```
<footer>
  <p>Posted by: W3School</p>
  <p>Contact information: <a href="mailto:someone@example.com">someone@example.com</a>.</
</footer>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9, Firefox, Opera, Chrome, and Safari 支持 <footer> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <footer> 标签。

定义和用法

<footer> 标签定义文档或节的页脚。

<footer> 元素应当含有其包含元素的信息。

页脚通常包含文档的作者、版权信息、使用条款链接、联系信息等等。

您可以在一个文档中使用多个 <footer> 元素。

HTML 4.01 与 HTML 5 之间的差异

<footer> 标签是 HTML 5 中的新标签。

提示和注释

提示：<footer> 元素内的联系信息应该位于 <address> 标签中。

全局属性

<footer> 标签支持 HTML 中的全局属性。

事件属性

<footer> 标签支持 HTML 中的事件属性。

HTML <form> 标签

例子

```
<form action="form_action.asp" method="get">
  <p>First name: <input type="text" name="fname" /></p>
  <p>Last name: <input type="text" name="lname" /></p>
  <input type="submit" value="Submit" />
</form>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <form> 标签。

定义和用法

<form> 标签用于为用户输入创建 HTML 表单。

表单能够包含 input 元素，比如文本字段、复选框、单选框、提交按钮等等。

表单还可以包含 menus、textarea、fieldset、legend 和 label 元素。

表单用于向服务器传输数据。

提示和注释

注释：form 元素是块级元素，其前后会产生折行。

HTML 与 XHTML 之间的差异

NONE

属性

new : HTML5 中的新属性。

属性	值	描述
accept	*MIME_type*	HTML 5 中不支持。
accept-charset	*charset_list*	规定服务器可处理的表单数据字符集。
action	*URL*	规定当提交表单时向何处发送表单数据。
autocomplete	`on` `off`	规定是否启用表单的自动完成功能。
enctype	见说明	规定在发送表单数据之前如何对其进行编码。
method	`get` `post`	规定用于发送 form-data 的 HTTP 方法。
name	*form_name*	规定表单的名称。
novalidate	novalidate	如果使用该属性，则提交表单时不进行验证。
target	`_blank` `_self` `_parent` `_top` `framename`	规定在何处打开 action URL。

说明

enctype 属性可能的值：

- application/x-www-form-urlencoded
- multipart/form-data
- text/plain

全局属性

<form> 标签支持 HTML 中的全局属性。

事件属性

<form> 标签支持 HTML 中的事件属性。

TIY 实例

文本域(Text fields)

本例演示如何在HTML页面创建文本域。用户可以在文本域写入文本。

```
<html>

<body>

<form>
名：
<input type="text" name="firstname">
<br />
姓：
<input type="text" name="lastname">
</form>

</body>
</html>
```

密码域

本例演示如何创建HTML的密码域。

```
<html>

<body>

<form>
用户：
<input type="text" name="user">
<br />
密码：
<input type="password" name="password">
</form>
<p>
请注意，当您在密码域中键入字符时，浏览器将使用项目符号来代替这些字符。
</p>
</body>
</html>
```

复选框

本例演示如何在HTML页中创建文本框。用户可以选中或取消选取复选框。

```
<html>

<body>

<form>
我喜欢自行车：
<input type="checkbox" name="Bike">
<br />
我喜欢汽车：
<input type="checkbox" name="Car">
</form>

</body>
</html>
```

单选按钮

本例演示如何在HTML中创建单选按钮。

```
<html>

<body>

<form>
男性：
<input type="radio" checked="checked" name="Sex" value="male" />
<br />
女性：
<input type="radio" name="Sex" value="female" />
</form>

<p>当用户点击一个单选按钮时，该按钮会变为选中状态，其他所有按钮会变为非选中状态。</p>

</body>
</html>
```

文本域(Textarea)

本例演示如何创建一个文本域（多行文本输入控制）。用户可以在文本域中写入文本。在文本域中，可写入的字符字数不受限制。

```
<html>
<body>

<p>
This example cannot be edited
because our editor uses a textarea
for input,
and your browser does not allow
a textarea inside a textarea.
</p>

<textarea rows="10" cols="30">
The cat was playing in the garden.
```

创建按钮

本例演示如何创建按钮。你可以对按钮上的文字进行自定义。

```
<html>

<body>

<form>
<input type="button" value="Hello world!">
</form>

</body>
</html>
```

带有输入框和确认按钮的表单

本例演示如何向页面添加表单。此表单包含两个输入框和一个确认按钮。

```
<html>
<body>

<form action="/example/html/form_action.asp" method="get">
  <p>First name: <input type="text" name="fname" /></p>
  <p>Last name: <input type="text" name="lname" /></p>
  <input type="submit" value="Submit" />
</form>

<p>请单击确认按钮，输入会发送到服务器上名为 "form_action.asp" 的页面。</p>

</body>
</html>
```

带有复选框的表单

此表单包含两个复选框和一个确认按钮。

```
<html>

<body>

<form name="input" action="/html/html_form_action.asp" method="get">
I have a bike:
<input type="checkbox" name="vehicle" value="Bike" checked="checked" />
<br />
I have a car:
<input type="checkbox" name="vehicle" value="Car" />
<br />
I have an airplane:
<input type="checkbox" name="vehicle" value="Airplane" />
<br /><br />
<input type="submit" value="Submit" />
</form>

<p>如果您点击 "Submit" 按钮，您将把输入传送到名为 html_form_action.asp 的新页面。</p>

</body>
</html>
```

带有单选按钮的表单

此表单包含两个单选框和一个确认按钮。

```
<html>

<body>

<form name="input" action="/html/html_form_action.asp" method="get">
Male:
<input type="radio" name="Sex" value="Male" checked="checked">
<br />
Female:
<input type="radio" name="Sex" value="Female">
<br />
<input type ="submit" value ="Submit">
</form>

<p>如果您点击 "Submit" 按钮，您将把输入传送到名为 html_form_action.asp 的新页面。</p>

</body>
</html>
```

从表单发送电子邮件

此例演示如何从表单发送电子邮件。

```
<html>

<body>
<form action="MAILTO:someone@w3school.com.cn" method="post" enctype="text/plain">

<h3>这个表单会把电子邮件发送到 W3School。</h3>
姓名：<br />
<input type="text" name="name" value="yourname" size="20">
<br />
电邮：<br />
<input type="text" name="mail" value="yourmail" size="20">
<br />
内容：<br />
<input type="text" name="comment" value="yourcomment" size="40">
<br /><br />
<input type="submit" value="发送">
<input type="reset" value="重置">

</form>
</body>
</html>
```

相关页面

HTML DOM 参考手册：Form 对象

HTML <frame> 标签

实例

简单的三框架页面：

```
<html>

<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
  <frame src="frame_c.htm" />
</frameset>

</html>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <frame> 标签。

定义和用法

<frame> 标签定义 frameset 中的一个特定的窗口（框架）。

frameset 中的每个框架都可以设置不同的属性，比如 border、scrolling、noresize 等等。

HTML 与 XHTML 之间的差异

在 HTML 中，<frame> 标签没有结束标签。

在 XHTML 中，<frame> 标签必须被正确地关闭。

提示和注释：

注释：如果您希望验证包含框架的页面，请确保 doctype 被设置为 "Frameset DTD"。阅读更多有关 DOCTYPE 的内容。

重要事项：您不能与 <frameset></frameset> 标签一起使用 <body></body> 标签。不过，如果您需要为不支持框架的浏览器添加一个 <noframes> 标签，请务必将此标签放置在 <body></body> 标签中！

可选的属性

属性	值	描述
frameborder	`0` `1`	规定是否显示框架周围的边框。
longdesc	*URL*	规定一个包含有关框架内容的长描述的页面。
marginheight	*pixels*	定义框架的上方和下方的边距。
marginwidth	*pixels*	定义框架的左侧和右侧的边距。
name	*name*	规定框架的名称。
noresize	noresize	规定无法调整框架的大小。
scrolling	`yes` `no` `auto`	规定是否在框架中显示滚动条。
src	*URL*	规定在框架中显示的文档的 URL。

标准属性

```
id, class, title, style
```

如需完整的描述，请访问标准属性。

TIY 实例

垂直框架

本例演示：如何使用三份不同的文档制作一个垂直框架。

```
<html>

<frameset cols="25%,50%,25%">

  <frame src="/example/html/frame_a.html">
  <frame src="/example/html/frame_b.html">
  <frame src="/example/html/frame_c.html">

</frameset>

</html>
```

水平框架

本例演示：如何使用三份不同的文档制作一个水平框架。

```
<html>

<frameset rows="25%,50%,25%">

  <frame src="/example/html/frame_a.html">
  <frame src="/example/html/frame_b.html">
  <frame src="/example/html/frame_c.html">

</frameset>

</html>
```

如何使用 <noframes> 标签

本例演示：如何使用 <noframes> 标签。

```
<html>

<frameset cols="25%,50%,25%">
  <frame src="/example/html/frame_a.html">
  <frame src="/example/html/frame_b.html">
  <frame src="/example/html/frame_c.html">

<noframes>
<body>您的浏览器无法处理框架！</body>
</noframes>

</frameset>

</html>
```

混合框架结构

本例演示如何制作含有三份文档的框架结构，同时将他们混合置于行和列之中。

```
<html>

<frameset rows="50%,50%">

<frame src="/example/html/frame_a.html">

<frameset cols="25%,75%">
<frame src="/example/html/frame_b.html">
<frame src="/example/html/frame_c.html">
</frameset>

</frameset>

</html>
```

含有 noresize="noresize" 属性的框架结构

本例演示 noresize 属性。在本例中，框架是不可调整尺寸的。在框架间的边框上拖动鼠标，你会发现边框是无法移动的。

```
<html>

<frameset cols="50%,*,25%">
  <frame src="/example/html/frame_a.html" noresize="noresize" />
  <frame src="/example/html/frame_b.html" />
  <frame src="/example/html/frame_c.html" />
</frameset>

</html>
```

导航框架

本例演示如何制作导航框架。导航框架包含一个将第二个框架作为目标的链接列表。名为 "contents.htm" 的文件包含三个链接。

```
<html>

<frameset cols="120,*">

  <frame src="/example/html/html_contents.html">
  <frame src="/example/html/frame_a.html" name="showframe">

</frameset>

</html>
```

内联框架

本例演示如何创建内联框架（HTML 页中的框架）。

```
<html>

<body>

<iframe src="/i/eg_landscape.jpg"></iframe>

<p>一些老的浏览器不支持 iframe。</p>
<p>如果得不到支持，iframe 是不可见的。</p>

</body>
</html>
```

跳转至框架内的一个指定的节

本例演示两个框架。其中的一个框架设置了指向另一个文件内指定的节的链接。这个 "link.htm" 文件内指定的节使用 <a name="C10"> 进行标识。

```
<html>

<frameset cols="20%,80%">

 <frame src="/example/html/frame_a.html">
 <frame src="/example/html/link.html#C10">

</frameset>

</html>
```

使用框架导航跳转至指定的节

本利演示两个框架。左侧的导航框架包含了一个链接列表，这些链接将第二个框架作为目标。第二个框架显示被链接的文档。导航框架其中的链接指向目标文件中指定的节。

```
<html>

<frameset cols="180,*">

<frame src="/example/html/content.html">
<frame src="/example/html/link.html" name="showframe">

</frameset>

</html>
```

HTML <frameset> 标签

实例

简单的三框架页面：

```
<html>

<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
  <frame src="frame_c.htm" />
</frameset>

</html>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <frameset> 标签。

定义和用法

frameset 元素可定义一个框架集。它被用来组织多个窗口（框架）。每个框架存有独立的文档。在其最简单的应用中，frameset 元素仅仅会规定在框架集中存在多少列或多少行。您必须使用 cols 或 rows 属性。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：如果您希望验证某个包含框架的页面，请确保 DTD 被设置为 "Frameset DTD"。阅读更多有关 XHTML 验证的内容。

重要事项：您不能与 <frameset></frameset> 标签一起使用 <body></body> 标签。不过，如果您需要为不支持框架的浏览器添加一个 <noframes> 标签，请务必将此标签放置在 <body></body> 标签中！

可选的属性

属性	值	描述
cols	`pixels` `%` `*`	定义框架集中列的数目和尺寸。有关 cols 属性的详细信息。
rows	`pixels` `%` `*`	定义框架集中行的数目和尺寸。有关 rows 属性的详细信息。

标准属性

```
id, class, title, style
```

如需完整的描述，请访问标准属性。

事件属性

```
NONE
```

如需完整的描述，请访问事件属性。

HTML <h1> 到 <h6> 标签

实例

六个不同的 HTML 标题：

```
<h1>这是标题 1</h1>
<h2>这是标题 2</h2>
<h3>这是标题 3</h3>
<h4>这是标题 4</h4>
<h5>这是标题 5</h5>
<h6>这是标题 6</h6>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <h1> - <h6> 标签。

定义和用法

<h1> - <h6> 标签可定义标题。<h1> 定义最大的标题。<h6> 定义最小的标题。

由于 h 元素拥有确切的语义，因此请您慎重地选择恰当的标签层级来构建文档的结构。因此，请不要利用标题标签来改变同一行中的字体大小。相反，我们应当使用层叠样式表定义来达到漂亮的显示效果。

如果您希望了解更多有关 HTML 标签选择和使用的信息，请阅读《Web 品质》。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，h1 - h6 元素的 "align" 属性不被推荐使用。

在 XHTML 1.0 Strict DTD 中，h1 - h6 元素的 "align" 属性不被支持。

可选的属性

属性	值	描述
align	left center right justify	不推荐使用。请使用样式替代它。规定标题中文本的排列。

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

如需完整的描述，请访问事件属性。

TIY 实例

标题

本例演示在 HTML 文档中显示标题的标签。

```
<html>

<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

<p>请仅仅把标题标签用于标题文本。不要仅仅为了产生粗体文本而使用它们。请使用其它标签或 CSS 代替。</p>

</body>
</html>
```

居中排列的标题

本例演示一个居中排列的标题。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

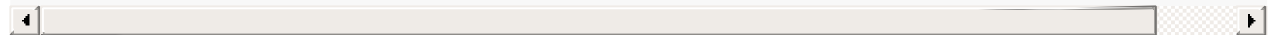
<html>

<body>

<h1 align="center">This is heading 1</h1>

<p>上面的标题在页面中进行了居中排列。上面的标题在页面中进行了居中排列。上面的标题在页面中进行了居中排列。<

</body>
</html>
```



HTML <head> 标签

实例

一个简单的 HTML 文档，带有最基本的必需的元素：

```
<html>

<head>
  <title>文档的标题</title>
</head>

<body>
  文档的内容... ...
</body>

</html>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <head> 标签。

定义和用法

<head> 标签用于定义文档的头部，它是所有头部元素的容器。<head> 中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。

文档的头部描述了文档的各种属性和信息，包括文档的标题、在 Web 中的位置以及和其他文档的关系等。绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

下面这些标签可用在 head 部分：<base>, <link>, <meta>, <script>, <style>, 以及 <title>。

<title> 定义文档的标题，它是 head 部分中唯一必需的元素。

提示和注释

提示：应该把 <head> 标签放在文档的开始处，紧跟在 <html> 后面，并处于 <body> 标签或 <frameset> 标签之前。

提示：请记住始终为文档规定标题！

HTML 与 XHTML 之间的差异

NONE

可选的属性

属性	值	描述
profile	*URL*	一个由空格分隔的 URL 列表，这些 URL 包含着有关页面的元数据信息。

profile 属性的更多信息

文档的头部经常会包含一些 <meta> 标签，用来告诉浏览器关于文档的附加信息。在将来，创作者可能会利用预先定义好的标准文档的元数据配置文件（metadata profile），以便更好地描述它们的文档。profile 属性提供了与当前文档相关联的配置文件的 URL。

配置文件的格式以及浏览器使用它们的方式都还没有进行定义，这个属性主要是为将来的开发而保留的占位符。

全局属性

<head> 标签支持 HTML 中的全局属性。

TIY 实例

文档的标题

头元素内部的标题信息不会被显示在浏览器窗口中。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<meta http-equiv="Content-Language" content="zh-cn" />

<title>标题不会显示在文档区</title>
</head>

<body>
<p>这段文本会显示出来。</p>
</body>

</html>
```

一个 target，所有的链接

本例显示如何使用 base 标签使页面中的所有标签在新窗口中打开。

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<meta http-equiv="Content-Language" content="zh-cn" />

<base target="_blank" />
</head>

<body>

<p>
<a href="http://www.w3school.com.cn" target="_blank">这个连接</a> 将在新窗口中加载，因为 targe
</p>

<p>
<a href="http://www.w3school.com.cn">这个连接</a> 也将在新窗口中加载，即使没有 target 属性。
</p>

</body>
</html>
```

HTML 中的样式

本例演示如何使用添加到 <head> 部分的样式信息对 HTML 进行格式化。

```
<html>

<head>
<style type="text/css">
h1 {color: red}
p {color: blue}
</style>
</head>

<body>
<h1>header 1</h1>
<p>A paragraph.</p>
</body>

</html>
```

链接到一个外部样式表

本例演示如何通过 <link> 标签链接到一个外部样式表。

```
<html>

<head>
<link rel="stylesheet" type="text/css" href="/html/csstest1.css" >
</head>

<body>
<h1>我通过外部样式表进行格式化。</h1>
<p>我也一样！</p>
</body>

</html>
```

文档描述

Meta 元素中的信息可以描述 HTML 文档。

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />

<meta name="author"
content="w3school.com.cn">

<meta name="revised"
content="David Yang,8/1/07">

<meta name="generator"
content="Dreamweaver 8.0en">

</head>

<body>
<p>本文档的 meta 属性标识了创作者和编辑软件。</p>
</body>

</html>
```

文档关键字

Meta 元素中的信息可以描述文档的关键词。

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />

<meta name="description"
content="HTML examples">

<meta name="keywords"
content="HTML, DHTML, CSS, XML, XHTML, JavaScript, VBScript">

</head>

<body>
<p>本文档的 meta 属性描述了该文档和它的关键词。</p>
</body>

</html>
```

重定向

这个例子演示：在网址已经变更的情况下，将用户重定向到另外一个地址。

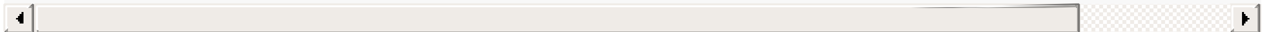
```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<meta http-equiv="Refresh" content="5;url=http://www.w3school.com.cn" />
</head>

<body>
<p>
对不起。我们已经搬家了。您的 URL 是 <a href="http://www.w3school.com.cn">http://www.w3school.c
</p>

<p>您将在 5 秒内被重定向到新的地址。</p>

<p>如果超过 5 秒后您仍然看到本消息，请点击上面的链接。</p>

</body>
</html>
```



HTML <header> 标签

实例

对主页的介绍：

```
<header>
<h1>Welcome to my homepage</h1>
<p>My name is Donald Duck</p>
</header>

<p>The rest of my home page...</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <header> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <header> 标签。

定义和用法

<header> 标签定义文档的页眉（介绍信息）。

HTML 4.01 与 HTML 5 之间的差异

<header> 标签是 HTML 5 中的新标签。

全局属性

<header> 标签支持 HTML 中的全局属性。

事件属性

<header> 标签支持 HTML 中的事件属性。

HTML <hr> 标签

实例

被水平线分隔的标题和段落：

```
<h1>This is header 1</h1>
<hr />
<p>This is some text</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <hr> 标签。

定义和用法

<hr> 标签在 HTML 页面中创建一条水平线。

水平分隔线（horizontal rule）可以在视觉上将文档分隔成各个部分。

HTML 与 XHTML 之间的差异

在 HTML 中，<hr> 标签没有结束标签。

在 XHTML 中，<hr> 必须被正确地关闭，比如 <hr />。

在 HTML 4.01 中，hr 元素的所有呈现属性均不被赞成使用。

在 XHTML 1.0 Strict DTD 中，hr 元素的所有呈现属性均不被支持。

可选的属性

属性	值	描述
align	`center` `left` `right`	不赞成使用。请使用样式取代它。规定 hr 元素的对齐方式。
noshade	noshade	不赞成使用。请使用样式取代它。规定 hr 元素的颜色呈现为纯色。
size	*pixels*	不赞成使用。请使用样式取代它。规定 hr 元素的高度（厚度）。
width	*pixels %*	不赞成使用。请使用样式取代它。规定 hr 元素的宽度。

全局属性

<hr> 标签支持 HTML 中的全局属性。

事件属性

<hr> 标签支持 HTML 中的事件属性。

TIY 实例

水平线

如何插入水平线。

```
<html>

<body>
<p>hr 标签定义水平线：</p>
<hr />
<p>这是段落。</p>
<hr />
<p>这是段落。</p>
<hr />
<p>这是段落。</p>
</body>
</html>
```

HTML <html> 标签

实例

```
<html>

<head>
  这里是文档的头部 ... ...
  ...
</head>

<body>
  这里是文档的主体 ... ...
  ...
</body>

</html>
```

定义和用法

此元素可告知浏览器其自身是一个 HTML 文档。

<html> 与 </html> 标签限定了文档的开始点和结束点，在它们之间是文档的头部和主体。正如您所了解的那样，文档的头部由 <head> 标签定义，而主体由 <body> 标签定义。

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <html> 标签。

HTML 与 XHTML 之间的差异

xmlns 属性在 XHTML 中是必需的，但在 HTML 中不是。不过，即使 XHTML 文档中的 <html> 没有使用此属性，W3C 的验证器也不会报错。这是因为 "xmlns=http://www.w3.org/1999/xhtml" 是一个固定值，即使您没有包含它，此值也会被添加到 <html> 标签中。

有关 xmlns 属性的更多信息。

提示和注释

注释：即使 html 元素是文档的根元素，它也不包含 doctype 元素。doctype 元素必须位于 html 元素之前。

属性

new : HTML5 中的新属性。

属性	值	描述
manifest	*url*	定义一个 URL，在这个 URL 上描述了文档的缓存信息。
xmlns	http://www.w3.org/1999/xhtml	定义 XML namespace 属性。

全局属性

<html> 标签支持 HTML 中的全局属性。

HTML <i> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <i> 标签。

定义和用法

<i> 标签显示斜体文本效果。

<i> 标签和基于内容的样式标签 <em> 类似。它告诉浏览器将包含其中的文本以斜体字（italic）或者倾斜（oblique）字体显示。如果这种斜体字对该浏览器不可用的话，可以使用高亮、反白或加下划线等样式。

提示：<i> 标签一定要和结束标签 </i> 结合起来使用。

参阅

请参阅 HTML 5 参考手册，了解有关 <i> 标签的最新信息：

HTML 5 <i> 标签

HTML <iframe> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <iframe> 标签。

定义和用法

iframe 元素会创建包含另外一个文档的内联框架（即行内框架）。

HTML 与 XHTML 之间的差异

在 HTML 4.1 Strict DTD 和 XHTML 1.0 Strict DTD 中，不支持 iframe 元素。

提示和注释：

提示：您可以把需要的文本放置在 <iframe> 和 </iframe> 之间，这样就可以应对无法理解 iframe 的浏览器。

属性

new : HTML5 中的新属性。

属性	值	描述
align	`left` `right` `top` `middle` `bottom`	不赞成使用。请使用样式代替。规定如何根据周围的元素来对齐此框架。
frameborder	`1` `0`	规定是否显示框架周围的边框。
height	*pixels %*	规定 iframe 的高度。
longdesc	*URL*	规定一个页面，该页面包含了有关 iframe 的较长描述。
marginheight	*pixels*	定义 iframe 的顶部和底部的边距。
marginwidth	*pixels*	定义 iframe 的左侧和右侧的边距。
name	*frame_name*	规定 iframe 的名称。
sandbox	`""` `allow-forms` `allow-same-origin` `allow-scripts` `allow-top-navigation`	启用一系列对 <iframe> 中内容的额外限制。
scrolling	`yes` `no` `auto`	规定是否在 iframe 中显示滚动条。
seamless	seamless	规定 <iframe> 看上去像是包含文档的一部分。
src	*URL*	规定在 iframe 中显示的文档的 URL。
srcdoc	*HTML_code*	规定在 <iframe> 中显示的页面的 HTML 内容。
width	*pixels %*	定义 iframe 的宽度。

全局属性

<iframe> 标签支持 HTML 中的全局属性。

事件属性

<iframe> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：IFrame 对象

HTML <img> 标签

实例

在下面的例子中，我们在页面中插入一幅 W3School 的工程师在上海鲜花港拍摄的郁金香照片：

```
<img src="/i/eg_tulip.jpg"  alt="上海鲜花港 - 郁金香" />
```

以上代码的效果：



（您可以在页面底部找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <img> 标签。

定义和用法

img 元素向网页中嵌入一幅图像。

请注意，从技术上讲，<img> 标签并不会在网页中插入图像，而是从网页上链接图像。<img> 标签创建的是被引用图像的占位空间。

<img> 标签有两个必需的属性：src 属性 和 alt 属性。

延伸阅读：如何正确地使用图像

HTML 与 XHTML 之间的差异

在 HTML 中，<img> 标签没有结束标签。

在 XHTML 中，<img> 标签必须被正确地关闭。

在 HTML 4.01 中，不推荐使用 image 元素的 "align"、"border"、"hspace" 以及 "vspace" 属性。

在 XHTML 1.0 Strict DTD 中，不支持 image 元素的 "align"、"border"、"hspace" 以及 "vspace" 属性。

必需的属性

属性	值	描述
alt	*text*	规定图像的替代文本。
src	*URL*	规定显示图像的 URL。

可选的属性

属性	值	描述
align	`top` `bottom` `middle` `left` `right`	不推荐使用。规定如何根据周围的文本来排列图像。
border	*pixels*	不推荐使用。定义图像周围的边框。
height	*pixels %*	定义图像的高度。
hspace	*pixels*	不推荐使用。定义图像左侧和右侧的空白。
ismap	*URL*	将图像定义为服务器端图像映射。
longdesc	*URL*	指向包含长的图像描述文档的 URL。
usemap	*URL*	将图像定义为客户器端图像映射。
vspace	*pixels*	不推荐使用。定义图像顶部和底部的空白。
width	*pixels %*	设置图像的宽度。

全局属性

<img> 标签支持 HTML 中的全局属性。

事件属性

<img> 标签支持 HTML 中的事件属性。

TIY 实例

插入图像

本例演示如何在网页中显示图像。

```
<!DOCTYPE HTML>
<html>

<body>

<p>
一幅图像：
<img src="/i/eg_mouse.jpg" width="128" height="128" />
</p>

<p>
一幅动画图像：
<img src="/i/eg_cute.gif" width="50" height="50" />
</p>

<p>请注意，插入动画图像的语法与插入普通图像的语法没有区别。</p>

</body>
</html>
```

从不同的位置插入图片

本例演示如何将其他文件夹或服务器的图片显示到网页中。

```
<html>

<body>

<p>
来自另一个文件夹的图像：
<img src="/i/ct_netscape.jpg" />
</p>

<p>
来自 W3School.com.cn 的图像：
<img src="http://www.w3school.com.cn/i/w3school_logo_white.gif" />
</p>

</body>
</html>
```

背景图片

本例演示如何添加背景图片到HTML页面。

```
<html>

<body background="/i/eg_background.jpg">

<h3>图像背景</h3>

<p>gif 和 jpg 文件均可用作 HTML 背景。</p>

<p>如果图像小于页面，图像会进行重复。</p>

</body>
</html>
```

排列图片

本例演示如何在文字中排列图像。

```
<html>

<body>

<h2>未设置对齐方式的图像：</h2>

<p>图像 <img src ="/i/eg_cute.gif"> 在文本中</p>

<h2>已设置对齐方式的图像：</h2>

<p>图像 <img src="/i/eg_cute.gif" align="bottom"> 在文本中</p>

<p>图像 <img src ="/i/eg_cute.gif" align="middle"> 在文本中</p>

<p>图像 <img src ="/i/eg_cute.gif" align="top"> 在文本中</p>

<p>请注意，bottom 对齐方式是默认的对齐方式。</p>

</body>
</html>
```

浮动图像

本例演示如何使图片浮动至段落的左边或右边。

```
<html>

<body>

<p>
<img src ="/i/eg_cute.gif" align ="left">
带有图像的一个段落。图像的 align 属性设置为 "left"。图像将浮动到文本的左侧。
</p>

<p>
<img src ="/i/eg_cute.gif" align ="right">
带有图像的一个段落。图像的 align 属性设置为 "right"。图像将浮动到文本的右侧。
</p>

</body>
</html>
```

调整图像尺寸

本例演示如何将图片调整到不同的尺寸。

```
<html>

<body>

<img src="/i/eg_mouse.jpg" width="50" height="50">

<br />

<img src="/i/eg_mouse.jpg" width="100" height="100">

<br />

<img src="/i/eg_mouse.jpg" width="200" height="200">

<p>通过改变 img 标签的 "height" 和 "width" 属性的值，您可以放大或缩小图像。</p>

</body>
</html>
```

为图片显示替换文本

本例演示如何为图片显示替换文本。在浏览器无法载入图像时，替换文本属性告诉读者她们失去的信息。为页面上的图像都加上替换文本属性是个好习惯。

```
<html>

<body>

<p>仅支持文本的浏览器无法显示图像，仅仅能够显示在图像的 "alt" 属性中指定的文本。在这里，"alt" 的文本是“

<p>请注意，如果您把鼠标指针移动到图像上，大多数浏览器会显示 "alt" 文本。</p>

<img src="/i/eg_goleft.gif" alt="向左转" />

<p>如果无法显示图像，将显示 "alt" 属性中的文本：</p>

<img src="/i/eg_goleft123.gif" alt="向左转" />

</body>
</html>
```

制作图像链接

本例演示如何将图像作为一个链接使用。

```
<html>

<body>
<p>
您也可以把图像作为链接来使用：
<a href="/example/html/lastpage.html">
<img border="0" src="/i/eg_buttonnext.gif" />
</a>
</p>

</body>
</html>
```

创建图像地图

本例显示如何创建带有可供点击区域的图像地图。其中的每个区域都是一个超级链接。

```
<html>
<body>

<p>请点击图像上的星球，把它们放大。</p>

<img
src="/i/eg_planets.jpg"
border="0" usemap="#planetmap"
alt="Planets" />

<map name="planetmap" id="planetmap">

<area
shape="circle"
coords="180,139,14"
href ="/example/html/venus.html"
target ="_blank"
alt="Venus" />

<area
shape="circle"
coords="129,161,10"
href ="/example/html/mercur.html"
target ="_blank"
alt="Mercury" />

<area
shape="rect"
coords="0,0,110,260"
href ="/example/html/sun.html"
target ="_blank"
alt="Sun" />

</map>

<p><b>注释：</b>img 元素中的 "usemap" 属性引用 map 元素中的 "id" 或 "name" 属性（根据浏览器），所以

</body>
</html>
```

延伸阅读 - 如何正确地使用图像

HTML 和 XHTML 最引人注目的特征之一，就是能够在文档的文本中包括图像，既可以把图像作为文档的内在对象（内联图像），也可以将其作为一个可通过超链接下载的单独文档，或者作为文档的背景。

合理地在文档内容中加入图像（静态的或者具有动画效果的图标、照片、说明、绘画，等等）时，会使文档变得更加生动活泼，更加引人入胜，而且看上去更加专业，更具信息性并易于浏览。还可以专门使一个图像成为超链接的可视引导图。

然而，如果过度使用图像，文档就会变得支离破碎，混乱不堪，而且无法阅读，用户在下载和查看该页面时，更会增加很多不必要的等待时间。

请阅读下面的文章，学习 Web 上的两种主要图像格式：GIF 和 JPEG，以及如何正确地使用图像：

- GIF 图像
- JPEG 图像
- 在 Web 上使用图像

参阅

- 教程：HTML 图像
- 参考手册：HTML 5 <img> 标签

相关页面

HTML DOM 参考手册：Image 对象

HTML <input> 标签

实例

一个简单的 HTML 表单，包含两个文本输入框和一个提交按钮：

```
<form action="form_action.asp" method="get">
  First name: <input type="text" name="fname" />
  Last name: <input type="text" name="lname" />
  <input type="submit" value="Submit" />
</form>
```

（您可以在页面底部找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <input> 标签。

定义和用法

<input> 标签用于搜集用户信息。

根据不同的 type 属性值，输入字段拥有很多种形式。输入字段可以是文本字段、复选框、掩码后的文本控件、单选按钮、按钮等等。

HTML 与 XHTML 之间的差异

在 HTML 中，<input> 标签没有结束标签。

在 XHTML 中，<input> 标签必须被正确地关闭。

提示和注释：

提示：请使用 label 元素为某个表单控件定义标签。

属性

new : HTML5 中的新属性。

属性	值	描述
accept	*mime_type*	规定通过文件上传来提交的文件的类型。
align	`left` `right` `top` `middle` `bottom`	不赞成使用。规定图像输入的对齐方式。
alt	text	定义图像输入的替代文本。
autocomplete	`on` `off`	规定是否使用输入字段的自动完成功能。
autofocus	autofocus	规定输入字段在页面加载时是否获得焦点。（不适用于 type="hidden"）
checked	checked	规定此 input 元素首次加载时应当被选中。
disabled	disabled	当 input 元素加载时禁用此元素。
form	*formname*	规定输入字段所属的一个或多个表单。
formaction	*URL*	覆盖表单的 action 属性。（适用于 type="submit" 和 type="image"）
formenctype	见注释	覆盖表单的 enctype 属性。（适用于 type="submit" 和 type="image"）
formmethod	`get` `post`	覆盖表单的 method 属性。（适用于 type="submit" 和 type="image"）
formnovalidate	formnovalidate	覆盖表单的 novalidate 属性。如果使用该属性，则提交表单时不进行验证。
formtarget	`_blank` `_self` `_parent` `_top` `framename`	覆盖表单的 target 属性。（适用于 type="submit" 和 type="image"）
height	*pixels %*	定义 input 字段的高度。（适用于 type="image"）
list	*datalist-id*	引用包含输入字段的预定义选项的 datalist 。
max	*number date*	规定输入字段的最大值。请与 "min" 属性配合使用，来创建合法值的范围。
maxlength	number	规定输入字段中的字符的最大长度。

min	*number date*	规定输入字段的最小值。请与 "max" 属性配合使用，来创建合法值的范围。
multiple	multiple	如果使用该属性，则允许一个以上的值。
name	field_name	定义 input 元素的名称。
pattern	*regexp_pattern*	规定输入字段的值的模式或格式。例如 pattern="[0-9]" 表示输入值必须是 0 与 9 之间的数字。
placeholder	*text*	规定帮助用户填写输入字段的提示。
readonly	readonly	规定输入字段为只读。
required	required	指示输入字段的值是必需的。
size	number_of_char	定义输入字段的宽度。
src	URL	定义以提交按钮形式显示的图像的 URL。
step	*number*	规定输入字的的合法数字间隔。
type	`button` `checkbox` `file` `hidden` `image` `password` `radio` `reset` `submit` `text`	规定 input 元素的类型。
value	*value*	规定 input 元素的值。
width	*pixels %*	定义 input 字段的宽度。（适用于 type="image"）

全局属性

<input> 标签支持 HTML 中的全局属性。

事件属性

<input> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：

- Input Button 对象
- Input Checkbox 对象

- Input Color 对象
- Input Date 对象
- Input Datetime 对象
- Input Datetime Local 对象
- Input Email 对象
- Input FileUpload 对象
- Input Hidden 对象
- Input Input Image 对象
- Input Month 对象
- Input Number 对象
- Input Password 对象
- Input Range 对象
- Input Radio 对象
- Input Reset 对象
- Input Input Search 对象
- Input Submit 对象
- Input Text 对象
- Input Input Time 对象
- Input Input URL 对象
- Input Week 对象

HTML DOM Button 对象

Button 对象

Button 对象代表 HTML 文档中的一个按钮。

该元素没有默认的行为，但是必须有一个 onclick 事件句柄以便使用。

在 HTML 文档中 <input type="button"> 标签每出现一次，一个 Button 对象 就会被创建。

您可以通过遍历表单的 elements[] 数组来访问某个按钮，或者通过使用 document.getElementById()。

Button 对象的属性

属性	描述
accessKey	设置或返回访问按钮的快捷键。
alt	设置或返回当浏览器无法显示按钮时供显示的替代文本。
disabled	设置或返回是否禁用按钮。
form	返回对包含该按钮的表单对象的引用。
id	设置或返回按钮的 id。
name	设置或返回按钮的名称。
tabIndex	设置或返回按钮的 tab 键控制次序。
type	返回按钮的表单元素类型。
value	设置或返回在按钮上显示的文本。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Button 对象的方法

方法	描述
blur()	把焦点从元素上移开。
click()	在某个按钮上模拟一次鼠标单击。
focus()	为某个按钮赋予焦点。

HTML DOM Checkbox 对象

Checkbox 对象

Checkbox 对象代表一个 HTML 表单中的 一个选择框。

在 HTML 文档中 <input type="checkbox"> 每出现一次，Checkbox 对象就会被创建。

您可以通过遍历表单的 elements[] 数组来访问某个选择框，或者通过使用 document.getElementById() 。

Checkbox 对象的属性

属性	描述
accessKey	设置或返回访问 checkbox 的快捷键。
alt	设置或返回不支持 checkbox 时显示的替代文本。
checked	设置或返回 checkbox 是否应被选中。
defaultChecked	返回 checked 属性的默认值。
disabled	设置或返回 checkbox 是否应被禁用。
form	返回对包含 checkbox 的表单的引用。
id	设置或返回 checkbox 的 id。
name	设置或返回 checkbox 的名称。
tabIndex	设置或返回 checkbox 的 tab 键控制次序。
type	返回 checkbox 的表单元素类型。
value	设置或返回 checkbox 的 value 属性的值

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Checkbox 对象的方法

方法	描述
blur()	从 checkbox 上移开焦点。
click()	模拟在 checkbox 中的一次鼠标点击。
focus()	为 checkbox 赋予焦点。

HTML DOM Color 对象

Color 对象

Color 对象是 HTML5 中的新对象。

Color 对象表示 HTML <input type="color"> 元素。

注释：Internet Explorer 和 Safari 不支持 <input type="color"> 元素。

访问 Color 对象

您可以通过使用 getElementById() 来访问 <color> 元素：

```
var x = document.getElementById("myColor");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Color 对象。

创建 Color 对象

您可以通过使用 document.createElement() 方法来创建 <color> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "color");
```

Color 对象属性

属性	描述
span	设置或返回列的 span 属性的值。
autocomplete	设置或返回色彩选择器的 autocomplete 属性值。
autofocus	设置或返回色彩选择器在页面加载后是否应自动获取焦点。
defaultValue	设置或返回色彩选择器的默认值。
disabled	设置或返回色彩选择器是否被禁用。
form	返回对包含色彩选择器的表单的引用。
list	返回对包含色彩选择器的 datalist 的引用。
name	设置或返回色彩选择器的 name 属性值。
type	返回色彩选择器的表单元素类型。
value	设置或返回色彩选择器的 value 属性值。

标准属性和事件

Color 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Input Date 对象

Input Date 对象

Input Date 对象是 HTML5 中的新对象。

Input Date 对象表示 HTML <input type="datetime"> 元素。

注释：Internet Explorer 或 Firefox 不支持 <input type="date"> 元素。

访问 Input Date 对象

您可以通过使用 getElementById() 来访问 <input type="date"> 元素：

```
var x = document.getElementById("myDate");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Input Date 对象。

创建 Input Date 对象

您可以通过使用 document.createElement() 方法来创建 <input type="date"> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "date");
```

Input Date 对象属性

属性	描述
autocomplete	设置或返回 date 字段的 autocomplete 属性值。
autofocus	设置或返回 date 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 date 字段的默认值。
disabled	设置或返回 date 字段是否被禁用。
form	返回对包含 date 字段的表单的引用。
list	返回对包含 date 字段的 datalist 的引用。
max	设置或返回 date 字段的 max 属性值。
min	设置或返回 date 字段的 min 属性值。
name	设置或返回 date 字段的 name 属性值。
readOnly	设置或返回 date 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 date 字段。
step	设置或返回 date 字段的 step 属性值。
type	返回 date 字段的表单元素类型。
value	设置或返回 date 字段的 value 属性值。

标准属性和事件

Input Date 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Datetime 对象

Datetime 对象

Datetime 对象是 HTML5 中的新对象。

Datetime 对象表示 HTML <input type="datetime"> 元素。

注释：Internet Explorer、Firefox 或者 Chrome 不支持 <input type="datetime"> 元素，Safari 中部分支持。Opera 12 以及更早的版本中完全支持。

访问 Datetime 对象

您可以通过使用 getElementById() 来访问 <input type="myDatetime"> 元素：

```
var x = document.getElementById("myDatetime");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Datetime 对象。

创建 Datetime 对象

您可以通过使用 document.createElement() 方法来创建 <input type="datetime"> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "datetime");
```

Datetime 对象属性

属性	描述
autocomplete	设置或返回 datetime 字段的 autocomplete 属性值。
autofocus	设置或返回 datetime 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 datetime 字段的默认值。
disabled	设置或返回 datetime 字段是否被禁用。
form	返回对包含 datetime 字段的表单的引用。
list	返回对包含 datetime 字段的 datalist 的引用。
max	设置或返回 datetime 字段的 max 属性值。
min	设置或返回 datetime 字段的 min 属性值。
name	设置或返回 datetime 字段的 name 属性值。
readOnly	设置或返回 datetime 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 datetime 字段。
step	设置或返回 datetime 字段的 step 属性值。
type	返回 datetime 字段的表单元素类型。
value	设置或返回 datetime 字段的 value 属性值。

标准属性和事件

Datetime 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Datetime Local 对象

Datetime Local 对象

Datetime Local 对象是 HTML5 中的新对象。

Datetime Local 对象表示 HTML <colgroup> 元素。

注释：Internet Explorer 或 Firefox 不支持 <input type="datetime-local"> 元素。

访问 Datetime Local 对象

您可以通过使用 getElementById() 来访问 <datetime-local> 元素：

```
var x = document.getElementById("myLocalDate");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Input Date 对象。

创建 Datetime Local 对象

您可以通过使用 document.createElement() 方法来创建 <datetime-local> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "datetime-local");
```

Datetime Local 对象属性

属性	描述
autocomplete	设置或返回本地时间字段的 autocomplete 属性值。
autofocus	设置或返回本地时间字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回本地时间字段的默认值。
disabled	设置或返回本地时间字段是否被禁用。
form	返回对包含本地时间字段的表单的引用。
list	返回对包含本地时间字段的 datalist 的引用。
max	设置或返回本地时间字段的 max 属性值。
min	设置或返回本地时间字段的 min 属性值。
name	设置或返回本地时间字段的 name 属性值。
readOnly	设置或返回本地时间字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写本地时间字段。
step	设置或返回本地时间字段的 step 属性值。
type	返回本地时间字段的表单元素类型。
value	设置或返回本地时间字段的 value 属性值。

标准属性和事件

Datetime Local 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Email 对象

Email 对象

Email 对象是 HTML5 中的新对象。

Email 对象表示 HTML <email> 元素。

注释：Internet Explorer 9（以及更早的版本）或 Safari 不支持 <input type="email"> 元素。

访问 Email 对象

您可以通过使用 getElementById() 来访问 <email> 元素：

```
var x = document.getElementById("myEmail");
```

创建 Email 对象

您可以通过使用 document.createElement() 方法来创建 <email> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "email");
```

Email 对象属性

属性	描述
autocomplete	设置或返回 email 字段的 autocomplete 属性值。
autofocus	设置或返回 email 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 email 字段的默认值。
disabled	设置或返回 email 字段是否被禁用。
form	返回对包含 email 字段的表单的引用。
list	返回对包含 email 字段的 datalist 的引用。
max	设置或返回 email 字段的 max 属性值。
min	设置或返回 email 字段的 min 属性值。
name	设置或返回 email 字段的 name 属性值。
pattern	设置或返回 email 字段的 pattern 属性值。
placeholder	设置或返回 email 字段的 placeholder 属性值。
readOnly	设置或返回 email 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 email 字段。
size	设置或返回 email 字段的 size 属性值。
step	设置或返回 email 字段的 step 属性值。
type	返回 email 字段的表单元素类型。
value	设置或返回 email 字段的 value 属性值。

标准属性和事件

Email 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM FileUpload 对象

FileUpload 对象

在 HTML 文档中 <input type="file"> 标签每出现一次，一个 FileUpload 对象就会被创建。

该元素包含一个文本输入字段，用来输入文件名，还有一个按钮，用来打开文件选择对话框以便图形化选择文件。

该元素的 value 属性保存了用户指定的文件的名称，但是当包含一个 file-upload 元素的表单被提交的时候，浏览器会向服务器发送选中的文件的内容而不仅仅是发送文件名。

为安全起见，file-upload 元素不允许 HTML 作者或 JavaScript 程序员指定一个默认的文件名。HTML value 属性被忽略，并且对于此类元素来说，value 属性是只读的，这意味着只有用户可以输入一个文件名。当用户选择或编辑一个文件名，file-upload 元素触发 onchange 事件句柄。

您可以通过遍历表单的 elements[] 数组，或者通过使用 document.getElementById()来访问 FileUpload 对象。

FileUpload 对象的属性

属性	描述
accept	设置或返回指示文件传输的 MIME 类型的列表（逗号分隔）。
accessKey	设置或返回访问 FileUpload 对象的快捷键。
alt	设置或返回不支持 <input type="file"> 时显示的替代文字。
defaultValue	设置或返回 FileUpload 对象的初始值。
disabled	设置或返回是否禁用 FileUpload 对象。
form	返回对包含 FileUpload 对象的表单的引用。
id	设置或返回 FileUpload 对象的 id。
name	设置或返回 FileUpload 对象的名称。
tabIndex	设置或返回定义 FileUpload 对象的 tab 键控制次序的索引号。
type	返回表单元素的类型。对于 FileUpload，则是 "file" 。
value	返回由用户输入设置的文本后，FileUpload 对象的文件名。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

FileUpload 对象的方法

方法	描述
blur()	从 FileUpload 对象上移开焦点。
focus()	为 FileUpload 对象赋予焦点。
select()	选取 FileUpload 对象。

HTML DOM Hidden 对象

Hidden 对象

Hidden 对象代表一个 HTML 表单中的某个隐藏输入域。

这种类型的输入元素实际上是隐藏的。这个不可见的表单元素的 value 属性保存了一个要提交给 Web 服务器的任意字符串。如果想要提交并非用户直接输入的数据的话，就是用这种类型的元素。

在 HTML 表单中 <input type="hidden"> 标签每出现一次，一个 Hidden 对象就会被创建。

您可通过遍历表单的 elements[] 数组来访问某个隐藏输入域，或者通过使用 document.getElementById()。

Hidden 对象的属性

属性	描述
alt	设置或返回当不支持隐藏输入域时显示的替代文本。
form	返回一个对包含隐藏域的表单的引用。
id	设置或返回隐藏域的 id。
name	设置或返回隐藏域的名称。
type	返回隐藏输入域的表单类型。
value	设置或返回隐藏域的 value 属性的值。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

HTML DOM Input Image 对象

Input Image 对象

Input Image 对象是 HTML5 中的新对象。

Input Image 对象表示 HTML <image> 元素。

访问 Input Image 对象

您可以通过使用 getElementById() 来访问 <image> 元素：

```
var x = document.getElementById("myImage");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Input Image 对象。

创建 Input Image 对象

您可以通过使用 document.createElement() 方法来创建 <image> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "image");
```

Input Image 对象属性

属性	描述
autocomplete	设置或返回 email 字段的 autocomplete 属性值。
alt	设置或返回 input image 的 alt 属性值。
autofocus	设置或返回 input image 在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 input image 的默认值。
disabled	设置或返回 input image 是否被禁用。
form	返回对包含 input image 的表单的引用。
formAction	设置或返回 input image 的 formaction 属性值。
formEnctype	设置或返回 input image 的 formenctype 属性值。
formMethod	设置或返回 input image 的 formmethod 属性值。
formNoValidate	设置或返回在提交表单时是否应该验证表单数据（form-data）。
formTarget	设置或返回 input image 的 formtarget 属性值。
height	设置或返回 input image 的 height 属性值。
name	设置或返回 input image 的 name 属性值。
src	设置或返回 input image 的 src 属性值。
type	返回 input image 字段的表单元素类型。
value	设置或返回 input image 的 value 属性值。
width	设置或返回 input image 的 width 属性值。

标准属性和事件

Input Image 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Month 对象

Month 对象

Month 对象是 HTML5 中的新对象。

Month 对象表示 HTML <input type="month"> 元素。

注释：Internet Explorer 或 Firefox 不支持 <input type="month"> 元素。

访问 Month 对象

您可以通过使用 getElementById() 来访问 <input type="month"> 元素：

```
var x = document.getElementById("myMonth");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Month 对象。

创建 Month 对象

您可以通过使用 document.createElement() 方法来创建 <input type="month"> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "month");
```

Month 对象属性

属性	描述
autocomplete	设置或返回 month 字段的 autocomplete 属性值。
autofocus	设置或返回 month 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 month 字段的默认值。
disabled	设置或返回 month 字段是否被禁用。
form	返回对包含 month 字段的表单的引用。
list	返回对包含 month 字段的 datalist 的引用。
max	设置或返回 month 字段的 max 属性值。
min	设置或返回 month 字段的 min 属性值。
name	设置或返回 month 字段的 name 属性值。
readOnly	设置或返回 month 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 month 字段。
step	设置或返回 month 字段的 step 属性值。
type	返回 month 字段的表单元素类型。
value	设置或返回 month 字段的 value 属性值。

标准属性和事件

Month 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Number 对象

Number 对象

Number 对象是 HTML5 中的新对象。

Number 对象表示 HTML <input type="number"> 元素。

注释：Internet Explorer 9 以及更早的版本不支持 <input type="number"> 元素。

访问 Number 对象

您可以通过使用 getElementById() 来访问 <number> 元素：

```
var x = document.getElementById("myNumber");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Number 对象。

创建 Number 对象

您可以通过使用 document.createElement() 方法来创建 <number> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "number");
```

Number 对象属性

属性	描述
autocomplete	设置或返回 number 字段的 autocomplete 属性值。
autofocus	设置或返回 number 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 number 字段的默认值。
disabled	设置或返回 number 字段是否被禁用。
form	返回对包含 number 字段的表单的引用。
list	返回对包含 number 字段的 datalist 的引用。
max	设置或返回 number 字段的 max 属性值。
min	设置或返回 number 字段的 min 属性值。
name	设置或返回 number 字段的 name 属性值。
placeholder	设置或返回 number 字段的 placeholder 属性值。
readOnly	设置或返回 number 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 number 字段。
step	设置或返回 number 字段的 step 属性值。
type	返回 number 字段的表单元素类型。
value	设置或返回 number 字段的 value 属性值。

标准属性和事件

Number 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Password 对象

Password 对象

Password 对象代表 HTML 表单中的密码字段。

HTML 的 <input type="password"> 标签在表单上每出现一次，一个 Password 对象就会被创建。

该文本输入字段供用户输入某些敏感的数据，比如密码等。当用户输入的时候，他的输入是被掩盖的（例如使用星号*），以防止旁边的人从他背后看到输入的内容。不过需要注意的是，当表单提交时，输入是用明文发送的。

与类型为 "text" 的元素类似，当用户改变显示值时，它会触发 onchange 事件句柄。

您可以通过遍历表单的 elements[] array 来访问密码字段，或者通过使用 document.getElementById() 。

Password 对象属性

属性	描述
accessKey	设置或返回访问密码字段的快捷键。
alt	设置或返回当不支持密码字段时显示的替代文字。
defaultValue	设置或返回密码字段的默认值。
disabled	设置或返回是否应被禁用密码字段。
form	返回对包含此密码字段的表单的引用。
id	设置或返回密码字段的 id。
maxLength	设置或返回密码字段中字符的最大数目。
name	设置或返回密码字段的名称。
readOnly	设置或返回密码字段是否应当是只读的。
size	设置或返回密码字段的长度。
tabIndex	设置或返回密码字段的 tab 键控制次序。
type	返回密码字段的表单元素类型。
value	设置或返回密码字段的 value 属性的值。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Password 对象方法

属性	描述
blur()	从密码字段移开焦点。
focus()	为密码字段赋予焦点。
select()	选取密码字段中的文本。

HTML DOM Input Range 对象

Input Range 对象

Input Range 对象是 HTML5 中的新对象。

Input Range 对象表示 HTML <input type="range"> 元素。

注释：Internet Explorer 9 以及更早的版本不支持 <input type="range"> 元素。

访问 Input Range 对象

您可以通过使用 getElementById() 来访问 <input type="range"> 元素：

```
var x = document.getElementById("myRange");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Input Range 对象。

创建 Input Range 对象

您可以通过使用 document.createElement() 方法来创建 <input type="range"> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "range");
```

Input Range 对象属性

属性	描述
autocomplete	设置或返回滑块控件的 autocomplete 属性值。
autofocus	设置或返回滑块控件在页面加载后是否应自动获取焦点。
defaultValue	设置或返回滑块控件的默认值。
disabled	设置或返回滑块控件是否被禁用。
form	返回对包含滑块控件的表单的引用。
list	返回对包含滑块控件的 datalist 的引用。
max	设置或返回滑块控件的 max 属性值。
min	设置或返回滑块控件的 min 属性值。
name	设置或返回滑块控件的 name 属性值。
step	设置或返回滑块控件的 step 属性值。
type	返回滑块控件的表单元素类型。
value	设置或返回滑块控件的 value 属性值。

标准属性和事件

Input Range 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Radio 对象

Radio 对象

Radio 对象代表 HTML 表单中的单选按钮。

在 HTML 表单中 <input type="radio"> 每出现一次，一个 Radio 对象就会被创建。

单选按钮是表示一组互斥选项按钮中的一个。当一个按钮被选中，之前选中的按钮就变为非选中的。

当单选按钮被选中或不选中时，该按钮就会触发 onclick 事件句柄。

您可通过遍历表单的 elements[] 数组来访问 Radio 对象，或者通过使用 document.getElementById()。

Radio 对象属性

属性	描述
accessKey	设置或返回访问单选按钮的快捷键。
alt	设置或返回在不支持单选按钮时显示的替代文本。
checked	设置或返回单选按钮的状态。
defaultChecked	返回单选按钮的默认状态。
disabled	设置或返回是否禁用单选按钮。
form	返回一个对包含此单选按钮的表单的引用。
id	设置或返回单选按钮的 id。
name	设置或返回单选按钮的名称。
tabIndex	设置或返回单选按钮的 tab 键控制次序。
type	返回单选按钮的表单类型。
value	设置或返回单选按钮的 value 属性的值。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Radio 对象方法

方法	描述
blur()	从单选按钮移开焦点。
click()	在单选按钮上模拟一次鼠标点击。
focus()	为单选按钮赋予焦点。

HTML DOM Reset 对象

Reset 对象

Reset 对象代表 HTML 表单中的一个重置按钮。

在 HTML 表单中 <input type="reset"> 标签每出现一次，一个 Reset 对象就会被创建。

当重置按钮被点击，包含它的表单中所有输入元素的值都重置为它们的默认值。默认值由 HTML value 属性或 JavaScript 的 defaultValue 属性指定。

重置按钮在重置表单之前触发 onclick 句柄，并且这个句柄可以通过返回 fasle 来取消。

参阅 Form.reset() 方法 和 Form.onreset 事件句柄。

您可以通过遍历表单的 elements[] 数组来访问某个重置按钮，或者通过使用 document.getElementById()。

Reset 对象属性

属性	描述
accesskey	设置或返回访问重置按钮的快捷键。
alt	设置或返回当浏览器不支持重置按钮时供显示的替代文本。
disabled	设置或返回重置按钮是否应被禁用。
form	返回一个对包含此重置按钮的表单对象的引用。
id	设置或返回重置按钮的 id。
name	设置或返回重置按钮的名称。
tabIndex	设置或返回重置按钮的 tab 键控制次序。
type	返回重置按钮的表单元素类型。
value	设置或返回重置按钮上显示的文本。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Reset 对象方法

方法	描述
blur()	从重置按钮上移开焦点。
click()	在重置按钮上模拟一次鼠标点击。
focus()	为重置按钮赋予焦点。

HTML DOM Input Search 对象

Input Search 对象

Input Search 对象是 HTML5 中的新对象。

Input Search 对象表示 HTML <input type="search"> 元素。

注释：Internet Explorer 9（以及更早的版本）或 Safari 不支持 <input type="search"> 元素。

访问 Input Search 对象

您可以通过使用 getElementById() 来访问 <search> 元素：

```
var x = document.getElementById("mySearch");
```

创建 Input Search 对象

您可以通过使用 document.createElement() 方法来创建 <search> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "search");
```

Input Search 对象属性

属性	描述
autocomplete	设置或返回 Search 字段的 autocomplete 属性值。
autofocus	设置或返回 Search 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 Search 字段的默认值。
disabled	设置或返回 Search 字段是否被禁用。
form	返回对包含 Search 字段的表单的引用。
list	返回对包含 Search 字段的 datalist 的引用。
maxLength	设置或返回 search 字段的 maxLength 属性值。
name	设置或返回 Search 字段的 name 属性值。
pattern	设置或返回 Search 字段的 pattern 属性值。
placeholder	设置或返回 Search 字段的 placeholder 属性值。
readOnly	设置或返回 Search 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 Search 字段。
size	设置或返回 Search 字段的 size 属性值。
step	设置或返回 Search 字段的 step 属性值。
type	返回 Search 字段的表单元素类型。
value	设置或返回 Search 字段的 value 属性值。

标准属性和事件

Input Search 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Submit 对象

Submit 对象

Submit 对象代表 HTML 表单中的一个提交按钮 (submit button)。

在 HTML 表单中 <input type="submit"> 标签每出现一次，一个 Submit 对象就会被创建。

在表单提交之前，触发 onclick 事件句柄，并且一个句柄可以通过返回 fasle 来取消表单提交。

参阅 Form.submit() 方法 和 Form.onsubmit 事件句柄。

实例：表单验证

您可以通过遍历表单的 elements[] 数组来访问某个提交按钮，或者通过使用 document.getElementById()。

Submit 对象属性

属性	描述
accessKey	设置或返回访问提交按钮的快捷键。
alt	设置或返回当浏览器不支持提交按钮时供显示的替代文本。
disabled	设置或返回提交按钮是否应被禁用。
form	返回一个对包含此提交按钮的表单的引用。
id	设置或返回提交按钮的 id。
name	设置或返回提交按钮的名称。
tabIndex	设置或返回提交按钮的 tab 键控制次序。
type	返回提交按钮的表单元素类型。
value	设置或返回在提交按钮上显示的文本。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Submit 对象方法

方法	描述
blur()	从提交按钮上移开焦点。
click()	在提交按钮上模拟一次鼠标点击。
focus()	为提交按钮赋予焦点。

HTML DOM Text 对象

Text 对象

Text 对象代表 HTML 表单中的文本输入域。

在 HTML 表单中 <input type="text"> 每出现一次，Text 对象就会被创建。

该元素可创建一个单行的文本输入字段。当用户编辑显示的文本并随后把输入焦点转移到其他元素的时候，会触发 onchange 事件句柄。

您可以使用 HTML <textarea> 标记来创建多行文本输入。参阅 Textarea 对象。

对于掩码文本输入，把 <input type="text"> 中的 type 设置为 "password"。参阅 Input Password。

您可以通过表单的 elements[] 数组来访问文本输入域，或者通过使用 document.getElementById()。

Text 对象属性

属性	描述
accessKey	设置或返回访问文本域的快捷键。
alt	设置或返回当浏览器不支持文本域时供显示的替代文本。
defaultValue	设置或返回文本域的默认值。
disabled	设置或返回文本域是否应被禁用。
form	返回一个对包含文本域的表单对象的引用。
id	设置或返回文本域的 id。
maxLength	设置或返回文本域中的最大字符数。
name	设置或返回文本域的名称。
readOnly	设置或返回文本域是否应是只读的。
size	设置或返回文本域的尺寸。
tabIndex	设置或返回文本域的 tab 键控制次序。
type	返回文本域的表单元素类型。
value	设置或返回文本域的 value 属性的值。

标准属性

属性	描述
className	设置或返回元素的 class 属性。
dir	设置或返回文本的方向。
lang	设置或返回元素的语言代码。
title	设置或返回元素的 title 属性。

Text 对象方法

方法	描述
blur()	从文本域上移开焦点。
focus()	在文本域上设置焦点。
select()	选取文本域中的内容。

HTML DOM Input Time 对象

Input Time 对象

Input Time 对象是 HTML5 中的新对象。

Input Time 对象表示 HTML <input type="time"> 元素。

注释：Internet Explorer 或 Firefox 不支持 <input type="time"> 元素。

访问 Input Time 对象

您可以通过使用 getElementById() 来访问 HTML <input type="time"> 元素：

```
var x = document.getElementById("myTime");
```

创建 Input Time 对象

您可以通过使用 document.createElement() 方法来创建HTML <input type="time"> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "time");
```

Input Time 对象属性

属性	描述
autocomplete	设置或返回 time 字段的 autocomplete 属性值。
autofocus	设置或返回 time 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 time 字段的默认值。
disabled	设置或返回 time 字段是否被禁用。
form	返回对包含 time 字段的表单的引用。
list	返回对包含 time 字段的 datalist 的引用。
max	设置或返回 time 字段的 max 属性值。
min	设置或返回 time 字段的 min 属性值。
name	设置或返回 time 字段的 name 属性值。
readOnly	设置或返回 time 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 time 字段。
size	设置或返回 time 字段的 size 属性值。
step	设置或返回 time 字段的 step 属性值。
type	返回 time 字段的表单元素类型。
value	设置或返回 time 字段的 value 属性值。

标准属性和事件

Input Time 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Input URL 对象

Input URL 对象

Input URL 对象是 HTML5 中的新对象。

Input URL 对象表示 HTML <input type="url"> 元素。

注释：Internet Explorer 9（以及更早的版本）或 Safari 不支持 <input type="search"> 元素。

访问 Input URL 对象

您可以通过使用 getElementById() 来访问 <input type="url"> 元素：

```
var x = document.getElementById("myUrl");
```

创建 Input URL 对象

您可以通过使用 document.createElement() 方法来创建 <input type="url"> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "url");
```

Input URL 对象属性

属性	描述
autocomplete	设置或返回 Search 字段的 autocomplete 属性值。
autofocus	设置或返回 Search 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 Search 字段的默认值。
disabled	设置或返回 Search 字段是否被禁用。
form	返回对包含 Search 字段的表单的引用。
list	返回对包含 Search 字段的 datalist 的引用。
maxLength	设置或返回 search 字段的 maxLength 属性值。
multiple	设置或返回是否允许用户在 URL 字段中输入不止一条 url 地址。
name	设置或返回 Search 字段的 name 属性值。
pattern	设置或返回 Search 字段的 pattern 属性值。
placeholder	设置或返回 Search 字段的 placeholder 属性值。
readOnly	设置或返回 Search 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 Search 字段。
size	设置或返回 Search 字段的 size 属性值。
step	设置或返回 Search 字段的 step 属性值。
type	返回 Search 字段的表单元素类型。
value	设置或返回 Search 字段的 value 属性值。

标准属性和事件

Input URL 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML DOM Input Week 对象

Input Week 对象

Input Week 对象是 HTML5 中的新对象。

Input Week 对象表示 HTML <input type="week"> 元素。

注释：Internet Explorer 或 Firefox 不支持 <input type="week"> 元素。

访问 Input Week 对象

您可以通过使用 getElementById() 来访问 <input type="week"> 元素：

```
var x = document.getElementById("myWeek");
```

提示：您也可以通过遍历表单的 elements 集合来访问 Input Week 对象。

创建 Input Week 对象

您可以通过使用 document.createElement() 方法来创建 <input type="week"> 元素：

```
var x = document.createElement("INPUT");
x.setAttribute("type", "week");
```

Input Week 对象属性

属性	描述
autocomplete	设置或返回 week 字段的 autocomplete 属性值。
autofocus	设置或返回 week 字段在页面加载后是否应自动获取焦点。
defaultValue	设置或返回 week 字段的默认值。
disabled	设置或返回 week 字段是否被禁用。
form	返回对包含 week 字段的表单的引用。
list	返回对包含 week 字段的 datalist 的引用。
max	设置或返回 week 字段的 max 属性值。
min	设置或返回 week 字段的 min 属性值。
name	设置或返回 week 字段的 name 属性值。
readOnly	设置或返回 week 字段是否是只读的。
required	设置或返回在提交表单之前是否必须填写 week 字段。
step	设置或返回 week 字段的 step 属性值。
type	返回 week 字段的表单元素类型。
value	设置或返回 week 字段的 value 属性值。

标准属性和事件

Input Week 对象支持标准属性和事件。

相关页面

HTML 教程：HTML 表单

HTML 参考手册：HTML <input> 标签

HTML 参考手册：HTML <input> type 属性

HTML <ins> 标签

实例

带有已删除部分和新插入部分的文本：

```
a dozen is <del>20</del> <ins>12</ins> pieces
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <ins> 标签。

注释：没有主流浏览器能够正确地显示 <ins> 标签的 cite 或 datetime 属性。

定义和用法

<ins> 标签定义已经被插入文档中的文本。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

提示：请与 <del> 一同使用，来描述文档中的更新和修正。

可选的属性

属性	值	描述
cite	URL	指向另外一个文档的 URL，此文档可解释文本被插入的原因。
datetime	YYYYMMDD	定义文本被插入的日期和时间。

全局属性

<ins> 标签支持 HTML 中的全局属性。

事件属性

<ins> 标签支持 HTML 中的事件属性。

TIY 实例

删除字效果和插入字效果

此例演示如何标示删除文本和插入文本。

```
<html>

<body>

<p>一打有 <del>二十</del> <ins>十二</ins> 件。</p>

<p>大多数浏览器会改写为删除文本和下划线文本。</p>

<p>一些老式的浏览器会把删除文本和下划线文本显示为普通文本。</p>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Ins 对象

HTML <keygen> 标签

实例

带有 keygen 字段的表单：

```
<form action="demo_keygen.asp" method="get">
Username: <input type="text" name="usr_name" />
Encryption: <keygen name="security" />
<input type="submit" />
</form>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Firefox, Chrome, Opera 以及 Safari 6 支持 <keygen> 标签。

定义和用法

<keygen> 标签规定用于表单的密钥对生成器字段。

当提交表单时，私钥存储在本地，公钥发送到服务器。

浏览器支持

所有主流浏览器都支持 <keygen> 标签，除了 Internet Explorer 和 Safari。

HTML 4.01 与 HTML 5 之间的差异

keygen 是 HTML 中的新元素。

属性

new : HTML5 中的新属性。

属性	值	描述
autofocus	autofocus	使 keygen 字段在页面加载时获得焦点。
challenge	challenge	如果使用，则将 keygen 的值设置为在提交时询问。
disabled	disabled	禁用 keytag 字段。
form	*formname*	定义该 keygen 字段所属的一个或多个表单。
keytype	rsa	定义 keytype。rsa 生成 RSA 密钥。
name	*fieldname*	定义 keygen 元素的唯一名称。name 属性用于在提交表单时搜集字段的值。

全局属性

<keygen> 标签支持 HTML 中的全局属性。

事件属性

<keygen> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Keygen 对象

HTML <label> 标签

实例

带有两个输入字段和相关标记的简单 HTML 表单：

```
<form>
  <label for="male">Male</label>
  <input type="radio" name="sex" id="male" />
  <br />
  <label for="female">Female</label>
  <input type="radio" name="sex" id="female" />
</form>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <label> 标签。

Safari 2 或更早的版本不支持 <label> 标签。

定义和用法

<label> 标签为 input 元素定义标注（标记）。

label 元素不会向用户呈现任何特殊效果。不过，它为鼠标用户改进了可用性。如果您在 label 元素内点击文本，就会触发此控件。就是说，当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上。

<label> 标签的 for 属性应当与相关元素的 id 属性相同。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释："for" 属性可把 label 绑定到另外一个元素。请把 "for" 属性的值设置为相关元素的 id 属性的值。

属性

new : HTML5 中的新属性。

属性	值	描述
for	*id*	规定 label 绑定到哪个表单元素。
form	*formid*	规定 label 字段所属的一个或多个表单。

全局属性

<label> 标签支持 HTML 中的全局属性。

事件属性

<label> 标签支持 HTML 中的事件属性。

TIY 实例

label

如何为控件定义标记。

```
<html>
<body>

<p>请点击文本标记之一，就可以触发相关控件：</p>

<form>
<label for="male">Male</label>
<input type="radio" name="sex" id="male" />
<br />
<label for="female">Female</label>
<input type="radio" name="sex" id="female" />
</form>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Column 对象

HTML <legend> 标签

实例

组合表单中的相关元素：

```
<form>
  <fieldset>
    <legend>health information</legend>
    height: <input type="text" />
    weight: <input type="text" />
  </fieldset>
</form>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <legend> 标签。

定义和用法

legend 元素为 fieldset 元素定义标题（caption）。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，align 属性不被赞成使用。

在 XHTML 1.0 Strict DTD 中，align 属性不被支持。

可选的属性

DTD 指示此属性允许在哪种 DTD 中使用。S=Strict, T=Transitional, F=Frameset.

属性	值	描述	DTD
align	top bottom left right	不赞成使用。请使用样式代替。为 fieldset 中的标题定义对齐方式。	TF

全局属性

<fieldset> 标签支持 HTML 中的全局属性。

事件属性

<fieldset> 标签支持 HTML 中的事件属性。

TIY 实例

围绕数据的 Fieldset

本例演示如何在数据周围绘制一个带标题的框。

```
<!DOCTYPE HTML>
<html>

<body>

<form>
  <fieldset>
    <legend>健康信息</legend>
    身高：<input type="text" />
    体重：<input type="text" />
  </fieldset>
</form>

<p>如果表单周围没有边框，说明您的浏览器太老了。</p>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Legend 对象

HTML <li> 标签

实例

```
<ol>
   <li>Coffee</li>
   <li>Tea</li>
   <li>Milk</li>
</ol>

<ul>
   <li>Coffee</li>
   <li>Tea</li>
   <li>Milk</li>
</ul>
```

（请在页面底部查看更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <li> 标签。

定义和用法

 标签定义列表项目。

 标签可用在有序列表 (<ol>) 和无序列表 (<ul>) 中。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，li 元素的 "type" 和 "value" 属性是不被赞成使用的。

在 XHTML 1.0 Strict DTD 中，li 元素的 "type" 和 "value" 属性是不被支持的。

提示和注释：

提示：请使用 CSS 来定义列表和列表项目的类型。

可选的属性

属性	值	描述
type	`A` `a` `I` `i` `1` `disc` `square` `circle`	不赞成使用。请使用样式取代它。规定使用哪种项目符号。
value	*number*	不赞成使用。请使用样式取代它。规定列表项目的数字。

全局属性

 标签支持 HTML 中的全局属性。

事件属性

 标签支持 HTML 中的事件属性。

TIY 实例

一个无序列表

本例演示一个无序列表。

```
<html>

<body>

<h4>一个无序列表：</h4>
<ul>
  <li>咖啡</li>
  <li>茶</li>
  <li>牛奶</li>
</ul>

</body>
</html>
```

不同类型的无序列表

本例演示不同类型的无序列表。

```
<html>
<body>

<h4>Disc 项目符号列表：</h4>
<ul type="disc">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>

<h4>Circle 项目符号列表：</h4>
<ul type="circle">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>

<h4>Square 项目符号列表：</h4>
<ul type="square">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>

</body>
</html>
```

一个有序列表

本例演示一个有序列表。

```
<!DOCTYPE html>
<html>
<body>

<ol>
  <li>咖啡</li>
  <li>牛奶</li>
  <li>茶</li>
</ol>

<ol start="50">
  <li>咖啡</li>
  <li>牛奶</li>
  <li>茶</li>
</ol>

</body>
</html>
```

不同类型的有序列表

本例演示不同类型的有序列表。

```
<html>
<body>

<h4>数字列表：</h4>
<ol>
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>

<h4>字母列表：</h4>
<ol type="A">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>

<h4>小写字母列表：</h4>
<ol type="a">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>

<h4>罗马字母列表：</h4>
<ol type="I">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>

<h4>小写罗马字母列表：</h4>
<ol type="i">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>

</body>
</html>
```

嵌套列表

本例演示如何嵌套列表。

```
<html>

<body>

<h4>一个嵌套列表：</h4>
<ul>
  <li>咖啡</li>
  <li>茶
    <ul>
    <li>红茶</li>
    <li>绿茶</li>
    </ul>
  </li>
  <li>牛奶</li>
</ul>

</body>
</html>
```

嵌套列表 2

本例演示更复杂的嵌套列表。

```
<html>

<body>

<h4>一个嵌套列表：</h4>
<ul>
  <li>咖啡</li>
  <li>茶
    <ul>
    <li>红茶</li>
    <li>绿茶
      <ul>
      <li>中国茶</li>
      <li>非洲茶</li>
      </ul>
    </li>
    </ul>
  </li>
  <li>牛奶</li>
</ul>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Li 对象

HTML <link> 标签

实例

链接一个外部样式表：

```
<head>
<link rel="stylesheet" type="text/css" href="theme.css" />
</head>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

在用于样式表时，<link> 标签得到了几乎所有浏览器的支持。但是几乎没有浏览器支持其他方面的用途。

定义和用法

<link> 标签定义文档与外部资源的关系。

<link> 标签最常见的用途是链接样式表。

HTML 与 XHTML 之间的差异

在 HTML 中，<link> 标签没有结束标签。

在 XHTML 中，<link> 标签必须被正确地关闭。

提示和注释：

注释：link 元素是空元素，它仅包含属性。

注释：此元素只能存在于 head 部分，不过它可出现任何次数。

属性

new : HTML5 中的新属性。

属性	值	描述
charset	*char_encoding*	HTML5 中不支持。
href	*URL*	规定被链接文档的位置。
hreflang	*language_code*	规定被链接文档中文本的语言。
media	*media_query*	规定被链接文档将被显示在什么设备上。
rel	`alternate` `author` `help` `icon` `licence` `next` `pingback` `prefetch` `prev` `search` `sidebar` `stylesheet` `tag`	规定当前文档与被链接文档之间的关系。
rev	*reversed relationship*	HTML5 中不支持。
sizes	*heightxwidth* any	规定被链接资源的尺寸。仅适用于 rel="icon"。
target	`_blank` `_self` `_top` `_parent` *`frame_name`*	HTML5 中不支持。
type	*MIME_type*	规定被链接文档的 MIME 类型。

全局属性

<link> 标签支持 HTML 中的全局属性。

事件属性

<link> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Link 对象

HTML <main> 标签

实例

```
<main>
  <h1>Web Browsers</h1>
  <p>Google Chrome、Firefox 以及 Internet Explorer 是目前最流行的浏览器。</p>

  <article>
    <h1>Google Chrome</h1>
    <p>Google Chrome 是由 Google 开发的一款免费的开源 web 浏览器，于 2008 年发布。</p>
  </article>

  <article>
    <h1>Internet Explorer</h1>
    <p>Internet Explorer 由微软开发的一款免费的 web 浏览器，发布于 1995 年。</p>
  </article>

  <article>
    <h1>Mozilla Firefox</h1>
    <p>Firefox 是一款来自 Mozilla 的免费开源 web 浏览器，发布于 2004 年。</p>
  </article>
</main>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <main> 标签，除了 Internet Explorer。

定义和用法

<main> 标签规定文档的主要内容。

<main> 元素中的内容对于文档来说应当是唯一的。它不应包含在文档中重复出现的内容，比如侧栏、导航栏、版权信息、站点标志或搜索表单。

注释：在一个文档中，不能出现一个以上的 <main> 元素。<main> 元素不能是以下元素的后代：<article>、<aside>、<footer>、<header> 或 <nav>。

HTML 4.01 与 HTML 5 之间的差异

<main> 标签是 HTML 5 中的新标签。

全局属性

<main> 标签支持 HTML 中的全局属性。

事件属性

<main> 标签支持 HTML 中的事件属性。

HTML <map> 标签

实例

带有可点击区域的图像映射：

```
<img src="planets.jpg" border="0" usemap="#planetmap" alt="Planets" />

<map name="planetmap" id="planetmap">
  <area shape="circle" coords="180,139,14" href ="venus.html" alt="Venus" />
  <area shape="circle" coords="129,161,10" href ="mercur.html" alt="Mercury" />
  <area shape="rect" coords="0,0,110,260" href ="sun.html" alt="Sun" />
</map>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <map> 标签。

定义和用法

定义一个客户端图像映射。图像映射（image-map）指带有可点击区域的一幅图像。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：area 元素永远嵌套在 map 元素内部。area 元素可定义图像映射中的区域。

注释：<img>中的 usemap 属性可引用 <map> 中的 id 或 name 属性（取决于浏览器），所以我们应同时向 <map> 添加 id 和 name 属性。

必需的属性

属性	值	描述
id	unique_name	为 map 标签定义唯一的名称。

可选的属性

属性	值	描述
name	*mapname*	为 image-map 规定的名称。

全局属性

<map> 标签支持 HTML 中的全局属性。

事件属性

<map> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Map 对象

HTML <mark> 标签

实例

突出显示部分文本：

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <mark> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <mark> 标签。

定义和用法

<mark> 标签定义带有记号的文本。请在需要突出显示文本时使用 <m> 标签。

HTML 4.01 与 HTML 5 之间的差异

<mark> 标签是 HTML 5 中的新标签。

全局属性

<mark> 标签支持 HTML 中的全局属性。

事件属性

<mark> 标签支持 HTML 中的事件属性。

HTML <menu> 标签

实例

带有两个菜单按钮 （"File" 和 "Edit"） 的工具栏，每个按钮都包含带有一系列选项的下拉列表：

```
<menu type="toolbar">
 <li>
 <menu label="File">
 <button type="button" onclick="file_new()">New...</button>
 <button type="button" onclick="file_open()">Open...</button>
 <button type="button" onclick="file_save()">Save</button>
 </menu>
 </li>
 <li>
 <menu label="Edit">
 <button type="button" onclick="edit_cut()">Cut</button>
 <button type="button" onclick="edit_copy()">Copy</button>
 <button type="button" onclick="edit_paste()">Paste</button>
 </menu>
 </li>
</menu>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前所有主流浏览器都不支持 <menu> 标签。

定义和用法

<menu> 标签定义命令的列表或菜单。

<menu> 标签用于上下文菜单、工具栏以及用于列出表单控件和命令。

提示和注释：

提示：请使用 CSS 来设置菜单列表的样式！

HTML 4.01 与 HTML5 之间的差异

在 HTML 4.01 中已弃用 <menu> 元素。

在 HTML5 中重新定义了 <menu> 元素。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，menu 元素不被赞成使用。

在 XHTML 1.0 Strict DTD 中，menu 元素不被支持。

属性

New: HTML5 中的新属性。

属性	值	描述
label	text	规定菜单的可见标签。
type	popup toolbar	规定要显示哪种菜单类型。

全局属性

<menu> 标签支持 HTML 中的全局属性。

事件属性

<menu> 标签支持 HTML 中的事件属性。

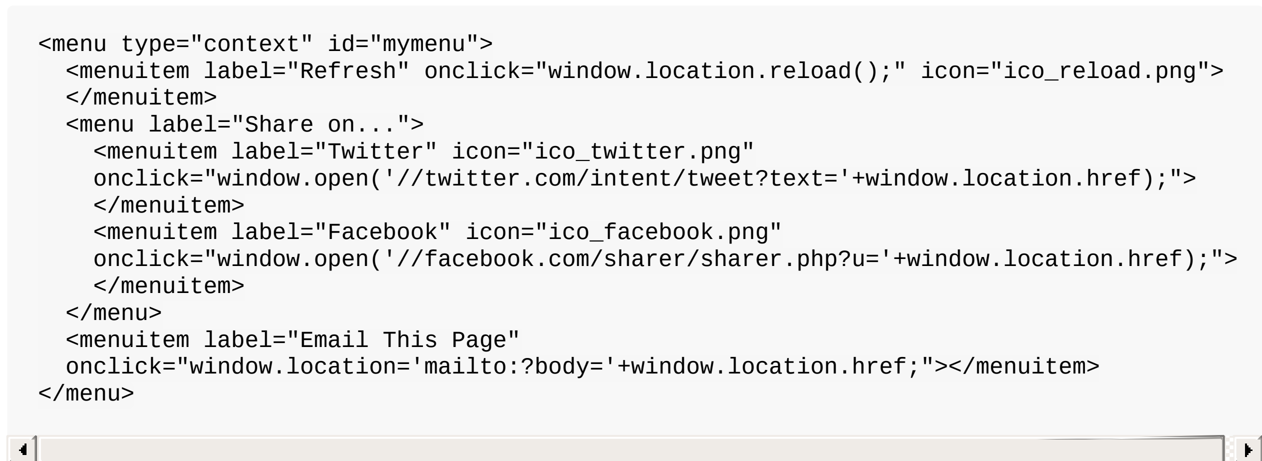
相关页面

HTML DOM 参考手册：Column 对象

HTML <menuitem> 标签

实例

包含不同 <menuitem> 元素的上下文菜单：



```
<menu type="context" id="mymenu">
  <menuitem label="Refresh" onclick="window.location.reload();" icon="ico_reload.png">
  </menuitem>
  <menu label="Share on...">
    <menuitem label="Twitter" icon="ico_twitter.png"
    onclick="window.open('//twitter.com/intent/tweet?text='+window.location.href);">
    </menuitem>
    <menuitem label="Facebook" icon="ico_facebook.png"
    onclick="window.open('//facebook.com/sharer/sharer.php?u='+window.location.href);">
    </menuitem>
  </menu>
  <menuitem label="Email This Page"
  onclick="window.location='mailto:?body='+window.location.href;"></menuitem>
</menu>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Firefox 8.0 以及更高的版本支持 <menuitem> 标签。

定义和用法

<menuitem> 标签定义用户可以从弹出菜单调用的命令/菜单项目。

HTML 4.01 与 HTML 5 之间的差异

<menuitem> 标签是 HTML5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
checked	checked	规定在页面加载后选中命令/菜单项目。仅适用于 type="radio" 或 type="checkbox"。
default	default	把命令/菜单项设置为默认命令。
disabled	disabled	规定命令/菜单项应该被禁用。
icon	URL	规定命令/菜单项的图标。
open	open	定义 details 是否可见。
label	*text*	必需。规定命令/菜单项的名称，以向用户显示。
radiogroup	*groupname*	规定命令组的名称，命令组会在命令/菜单项本身被切换时进行切换。仅适用于 type="radio"。

type	`checkbox`
	`command`
	`radio`
		规定命令/菜单项的类型。默认是 "command"。

全局属性

<menuitem> 标签支持 HTML 中的全局属性。

事件属性

<menuitem> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：MenuItem 对象

HTML <meta> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <meta> 标签。

定义和用法

<meta> 元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。

<meta> 标签位于文档的头部，不包含任何内容。<meta> 标签的属性定义了与文档相关联的名称/值对。

HTML 与 XHTML 之间的差异

在 HTML 中，<meta> 标签没有结束标签。

在 XHTML 中，<meta> 标签必须被正确地关闭。

提示和注释：

注释：<meta> 标签永远位于 head 元素内部。

注释：元数据总是以名称/值的形式被成对传递的。

必需的属性

属性	值	描述
content	some_text	定义与 http-equiv 或 name 属性相关的元信息

可选的属性

属性	值	描述
http-equiv	`content-type` `expires` `refresh` `set-cookie`	把 content 属性关联到 HTTP 头部。
name	`author` `description` `keywords` `generator` `revised` `others`	把 content 属性关联到一个名称。
scheme	some_text	定义用于翻译 content 属性值的格式。

name 属性

name 属性提供了名称/值对中的名称。HTML 和 XHTML 标签都没有指定任何预先定义的 <meta> 名称。通常情况下，您可以自由使用对自己和源文档的读者来说富有意义的名称。

"keywords" 是一个经常被用到的名称。它为文档定义了一组关键字。某些搜索引擎在遇到这些关键字时，会用这些关键字对文档进行分类。

类似这样的 meta 标签可能对于进入搜索引擎的索引有帮助：

```
<meta name="keywords" content="HTML,ASP,PHP,SQL">
```

如果没有提供 name 属性，那么名称/值对中的名称会采用 http-equiv 属性的值。

http-equiv 属性

http-equiv 属性为名称/值对提供了名称。并指示服务器在发送实际的文档之前先在要传送给浏览器的 MIME 文档头部包含名称/值对。

当服务器向浏览器发送文档时，会先发送许多名称/值对。虽然有些服务器会发送许多这种名称/值对，但是所有服务器都至少要发送一个：content-type:text/html。这将告诉浏览器准备接受一个 HTML 文档。

使用带有 http-equiv 属性的 <meta> 标签时，服务器将把名称/值对添加到发送给浏览器的内容头部。例如，添加：

```
<meta http-equiv="charset" content="iso-8859-1">
<meta http-equiv="expires" content="31 Dec 2008">
```

这样发送到浏览器的头部就应该包含：

```
content-type: text/html
charset:iso-8859-1
expires:31 Dec 2008
```

当然，只有浏览器可以接受这些附加的头部字段，并能以适当的方式使用它们时，这些字段才有意义。

content 属性

content 属性提供了名称/值对中的值。该值可以是任何有效的字符串。

content 属性始终要和 name 属性或 http-equiv 属性一起使用。

scheme 属性

scheme 属性用于指定要用来翻译属性值的方案。此方案应该在由 <head> 标签的 profile 属性指定的概况文件中进行了定义。

全局属性

<meta> 标签支持 HTML 中的全局属性。

相关页面

HTML DOM 参考手册：Meta 对象

HTML <meter> 标签

实例

使用 meter 元素来度量给定范围（gauge）内的数据：

```
<meter value="3" min="0" max="10">十分之三</meter>

<meter value="0.6">60%</meter>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Firefox, Chrome, Opera 以及 Safari 6 支持 <meter> 标签。

定义和用法

<meter> 标签定义已知范围或分数值内的标量测量。也被称为 gauge（尺度）。

例子：磁盘用量、查询结果的相关性，等等。

注释：<meter> 标签不应用于指示进度（在进度条中）。如果标记进度条，请使用 <progress> 标签。

HTML 4.01 与 HTML 5 之间的差异

<meter> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
form	*form_id*	规定 <meter> 元素所属的一个或多个表单。
high	*number*	规定被视作高的值的范围。
low	*number*	规定被视作低的值的范围。
max	*number*	规定范围的最大值。
min	*number*	规定范围的最小值。
optimum	*number*	规定度量的优化值。
value	*number*	必需。规定度量的当前值。

全局属性

<meter> 标签支持 HTML 中的全局属性。

事件属性

<meter> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Meter 对象

相关页面

HTML DOM 参考手册：Meter 对象

HTML <nav> 标签

实例

```
<nav>
<a href="index.asp">Home</a>
<a href="html5_meter.asp">Previous</a>
<a href="html5_noscript.asp">Next</a>
</nav>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9, Firefox, Opera, Chrome 以及 Safari 支持 <nav> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <nav> 标签。

定义和用法

<nav> 标签定义导航链接的部分。

HTML 4.01 与 HTML 5 之间的差异

<nav> 标签是 HTML 5 中的新标签。

提示和注释

提示：如果文档中有“前后”按钮，则应该把它放到 <nav> 元素中。

全局属性

<nav> 标签支持 HTML 中的全局属性。

事件属性

<nav> 标签支持 HTML 中的事件属性。

HTML <noframes> 标签

实例

```
<frameset cols = "25%, 25%,*">
  <noframes>
  <body>Your browser does not handle frames!</body>
  </noframes>
  <frame src ="venus.htm" />
  <frame src   ="sun.htm" />
  <frame src ="mercur.htm"   />
</frameset>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <noframes> 标签。

定义和用法

noframes 元素可为那些不支持框架的浏览器显示文本。noframes 元素位于 frameset 元素内部。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：如果浏览器有能力处理框架，就不会显示出 frameset 元素中的文本。

重要事项：如果您希望 frameset 添加 <noframes> 标签，就必须把其中的文本包装在 <body></body> 标签中！

注释：如果您希望验证包含框架的页面，请确保 DTD 被设置为 "Frameset DTD"。

阅读更多有关 XHTML 验证的内容。

注释：在XHTML 1.0 Strict DTD 中，<noframes> 标签是不被允许的。

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

TIY 实例

如何使用 <noframes> 标签

本例演示：如何使用 <noframes> 标签。

```
<html>

<frameset cols="25%,50%,25%">
  <frame src="/example/html/frame_a.html">
  <frame src="/example/html/frame_b.html">
  <frame src="/example/html/frame_c.html">

<noframes>
<body>您的浏览器无法处理框架！</body>
</noframes>

</frameset>

</html>
```

HTML <noscript> 标签

例子

JavaScript

```
<body>
  ...
  ...

  <script type="text/javascript">
    <!--
    document.write("Hello World!")
    //-->
  </script>

  <noscript>Your browser does not support JavaScript!</noscript>
  ...
  ...
</body>
```

VBScript

```
<body>
  ...
  ...
  <script type="text/vbscript">
   <!--
   document.write("Hello World!")
   '-->
  </script>

  <noscript>Your browser does not support VBScript!</noscript>
  ...
  ...
</body>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <noscript> 标签。

定义和用法

noscript 元素用来定义在脚本未被执行时的替代内容（文本）。

此标签可被用于可识别 <script> 标签但无法支持其中的脚本的浏览器。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：如果浏览器支持脚本，那么它不会显示出 noscript 元素中的文本。

注释：无法识别 <script> 标签的浏览器会把标签的内容显示到页面上。为了避免浏览器这样做，您应当在注释标签中隐藏脚本。老式的（无法识别 <script> 标签的）浏览器会忽略注释，这样就不会把标签的内容写到页面上，而新式的浏览器则懂得执行这些脚本，即使它们被包围在注释标签中！

全局属性

<noscript> 标签支持 HTML 中的全局属性。

HTML <object> 标签

实例

向 HTML 代码添加一个对象：

```
<object classid="clsid:F08DF954-8592-11D1-B16A-00C0F0283628" id="Slider1"
width="100" height="50">
  <param name="BorderStyle" value="1" />
  <param name="MousePointer" value="0" />
  <param name="Enabled" value="1" />
  <param name="Min" value="0" />
  <param name="Max" value="10" />
</object>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

几乎所有主流浏览器都拥有部分对 <object> 标签的支持。

定义和用法

定义一个嵌入的对象。请使用此元素向您的 XHTML 页面添加多媒体。此元素允许您规定插入 HTML 文档中的对象的数据和参数，以及可用来显示和操作数据的代码。

<object> 标签用于包含对象，比如图像、音频、视频、Java applets、ActiveX、PDF 以及 Flash。

object 的初衷是取代 img 和 applet 元素。不过由于漏洞以及缺乏浏览器支持，这一点并未实现。

浏览器的对象支持有赖于对象类型。不幸的是，主流浏览器都使用不同的代码来加载相同的对象类型。

而幸运的是，object 对象提供了解决方案。如果未显示 object 元素，就会执行位于 <object> 和 </object> 之间的代码。通过这种方式，我们能够嵌套多个 object 元素（每个对应一个浏览器）。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：<param> 标签定义用于对象的 run-time 设置。

注释：不要对图像使用 <object> 标签，请使用 <img> 标签代替。

可选的属性

属性	值	描述
align	`left` `right` `top` `bottom`	定义围绕该对象的文本对齐方式。
archive	*URL*	由空格分隔的指向档案文件的 URL 列表。这些档案文件包含了与对象相关的资源。
border	*pixels*	定义对象周围的边框。
classid	*class ID*	定义嵌入 Windows Registry 中或某个 URL 中的类的 ID 值，此属性可用来指定浏览器中包含的对象的位置，通常是一个 Java 类。
codebase	*URL*	定义在何处可找到对象所需的代码，提供一个基准 URL。
codetype	*MIME type*	通过 classid 属性所引用的代码的 MIME 类型。
data	*URL*	定义引用对象数据的 URL。如果有需要对象处理的数据文件,要用 data 属性来指定这些数据文件。
declare	declare	可定义此对象仅可被声明，但不能被创建或例示，直到此对象得到应用为止。
form	*form_id*	规定对象所属的一个或多个表单。
height	*pixels*	定义对象的高度。
hspace	*pixels*	定义对象周围水平方向的空白。
name	*unique_name*	为对象定义唯一的名称（以便在脚本中使用）。
standby	*text*	定义当对象正在加载时所显示的文本。
type	*MIME_type*	定义被规定在 data 属性中指定的文件中出现的数据的 MIME 类型。
usemap	*URL*	规定与对象一同使用的客户端图像映射的 URL。
vspace	*pixels*	定义对象的垂直方向的空白。
width	*pixels*	定义对象的宽度。

全局属性

<object> 标签支持 HTML 中的全局属性。

事件属性

<object> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Object 对象

HTML <ol> 标签

实例

有序 HTML 列表：

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <ol> 标签。

定义和用法

 标签定义有序列表。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，ol 元素的 "compact"、"start" 以及 "type" 属性是不被赞成使用的。

在 XHTML 1.0 Strict DTD 中，ol 元素的 "compact"、"start" 以及 "type" 属性是不被支持的。

提示和注释：

提示：请使用 CSS 来定义列表的类型。

属性

New：HTML5 中的新属性。

属性	值	描述
compact	compact	HTML5 中不支持。HTML 4.01 中不赞成使用。规定列表呈现的效果比正常情况更小巧。
reversed	reversed	规定列表顺序为降序。(9,8,7...)
start	*number*	规定有序列表的起始值。
type	`1` `A` `a` `I` `i`	规定在列表中使用的标记类型。

全局属性

 标签支持 HTML 中的全局属性。

事件属性

 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Ol 对象

HTML <optgroup> 标签

实例

通过 <optgroup> 标签把相关的选项组合在一起：

```
<select>
  <optgroup label="Swedish Cars">
    <option value ="volvo">Volvo</option>
    <option value ="saab">Saab</option>
  </optgroup>

  <optgroup label="German Cars">
    <option value ="mercedes">Mercedes</option>
    <option value ="audi">Audi</option>
  </optgroup>
</select>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <optgroup> 标签。

定义和用法

<optgroup> 标签定义选项组。

optgroup 元素用于组合选项。当您使用一个长的选项列表时，对相关的选项进行组合会使处理更加容易。

HTML 与 XHTML 之间的差异

NONE

必需的属性

属性	值	描述
label	*text*	为选项组规定描述。

可选的属性

属性	值	描述
disabled	disabled	规定禁用该选项组。

全局属性

<optgroup> 标签支持 HTML 中的全局属性。

事件属性

<optgroup> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：OptionGroup 对象

HTML <option> 标签

实例

创建带有 4 个选项的选择列表：

```
<select>
  <option value ="volvo">Volvo</option>
  <option value ="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <option> 标签。

定义和用法

option 元素定义下拉列表中的一个选项（一个条目）。

浏览器将 <option> 标签中的内容作为 <select> 标签的菜单或是滚动列表中的一个元素显示。

option 元素位于 select 元素内部。

HTML 与 XHTML 之间的差异

在 HTML 中，<option> 没有结束标签。

在 XHTML 中，<option> 必须被正确关闭。

提示和注释：

注释：<option> 标签可以在不带有任何属性的情况下使用，但是您通常需要使用 value 属性，此属性会指示出被送往服务器的内容。

注释：请与 select 元素配合使用此标签，否则这个标签是没有意义的。

提示：如果列表选项很多，可以使用 <optgroup> 标签对相关选项进行组合。

可选的属性

属性	值	描述
disabled	disabled	规定此选项应在首次加载时被禁用。
label	*text*	定义当使用 <optgroup> 时所使用的标注。
selected	selected	规定选项（在首次显示在列表中时）表现为选中状态。
value	*text*	定义送往服务器的选项值。

全局属性

<option> 标签支持 HTML 中的全局属性。

事件属性

<option> 标签支持 HTML 中的事件属性。

TIY 实例

简单的下拉列表

本例演示如何在 HTML 页面中创建简单的下拉列表框。下拉列表框是一个可选列表。

```
<html>

<body>

<form>
<select name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
</form>

</body>
</html>
```

另一个下拉列表

本例演示如何创建一个简单的带有预选值的下拉列表。（注：预选值指预先指定的首选项。）

```
<html>

<body>

<form>
<select name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat" selected="selected">Fiat</option>
<option value="audi">Audi</option>
</select>
</form>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Option 对象

HTML <output> 标签

实例

执行计算然后在 <output> 元素中显示结果：

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0
   <input type="range" id="a" value="50">100
   +<input type="number" id="b" value="50">
   =<output name="x" for="a b"></output>
</form>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Firefox, Chrome, Safari 以及 Opera 支持 <output> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <output> 标签。

定义和用法

<output> 标签定义不同类型的输出，比如脚本的输出。

HTML 4.01 与 HTML 5 之间的差异

<output> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
for	*element_id*	定义输出域相关的一个或多个元素。
form	*form_id*	定义输入字段所属的一个或多个表单。
name	*name*	定义对象的唯一名称。（表单提交时使用）

全局属性

<output> 标签支持 HTML 中的全局属性。

事件属性

<output> 标签支持 HTML 中的事件属性。

HTML <p> 标签

实例

以下代码标记了一个段落：

```
<p>This is some text in a very short paragraph</p>
```

（在页面下部，您可以找到更多实例。）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <p> 标签。

定义和用法

<p> 标签定义段落。

p 元素会自动在其前后创建一些空白。浏览器会自动添加这些空间，您也可以在样式表中规定。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，所有 p 元素的呈现属性均不被赞成使用。

在 XHTML 1.0 Strict DTD 中，所有 p 元素的呈现属性均不被支持。

可选的属性

属性	值	描述
align	`left` `right` `center` `justify`	不赞成使用。请使用样式取代它。规定段落中文本的对齐方式。

全局属性

<p> 标签支持 HTML 中的全局属性。

事件属性

<p> 标签支持 HTML 中的事件属性。

TIY 实例

简单的段落

此例演示：段落元素中的文字如何被浏览器显示。

```
<html>
<body>

<p>这是段落。</p>
<p>这是段落。</p>
<p>这是段落。</p>

<p>段落元素由 p 标签定义。</p>

</body>
</html>
```

更多的段落

本例演示段落元素的某些缺省的行为。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<body>

<p>
这个段落
在源代码中
包含许多行
但是浏览器
忽略了它们。
</p>

<p>
这个段落
在源代码          中
包含     许多行
但是          浏览器
忽略了    它们。
</p>

<p>
段落的行数依赖于浏览器窗口的大小。如果调节浏览器窗口的大小，将改变段落中的行数。
</p>

</body>
</html>
```

“诗歌”问题

本例演示 HTML 格式化的某些问题。

```
<html>

<body>

<h1>春晓</h1>

<p>
	春眠不觉晓，
		处处闻啼鸟。
			夜来风雨声，
				花落知多少。
</p>

<p>注意，浏览器忽略了源代码中的排版（省略了多余的空格和换行）。</p>

</body>

</html>
```

延伸阅读：允许的段落用法

可以只在块（block）内指定段落，也可以把段落和其他段落、列表、表单和预定义格式的文本一起使用。总的来讲，这意味着段落可以在任何有合适的文本流的地方出现，例如文档的主体中、列表的元素里，等等。

从技术角度将，段落不可以出现在头部、锚或者其他严格要求内容必须只能是文本的地方。实际上，多数浏览器都忽略了这个限制，它们会把段落作为所含元素的内容一起格式化。

HTML <param> 标签

实例

向 HTML 代码添加一个对象：

```
<object classid="clsid:F08DF954-8592-11D1-B16A-00C0F0283628" id="Slider1"
width="100" height="50">
  <param name="BorderStyle" value="1" />
  <param name="MousePointer" value="0" />
  <param name="Enabled" value="1" />
  <param name="Min" value="0" />
  <param name="Max" value="10" />
</object>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <param> 标签。

定义和用法

param 元素允许您为插入 XHTML 文档的对象规定 run-time 设置，也就是说，此标签可为包含它的 <object> 或者 <applet> 标签提供参数。

HTML 与 XHTML 之间的差异

在 HTML 中，<param> 标签没有结束标签。

在 XHTML 中，<param> 标签必须被正确地关闭。

必需的属性

属性	值	描述
name	*unique_name*	定义参数的名称（用在脚本中）。

可选的属性

属性	值	描述
type	*MIME type*	规定参数的 MIME 类型（internet media type）。
value	*value*	规定参数的值。
valuetype	`data` `ref` `object`	规定值的 MIME 类型。

全局属性

<param> 标签支持 HTML 中的全局属性。

事件属性

<param> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Parameter 对象

HTML <pre> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <pre> 标签。

定义和用法

pre 元素可定义预格式化的文本。被包围在 pre 元素中的文本通常会保留空格和换行符。而文本也会呈现为等宽字体。

<pre> 标签的一个常见应用就是用来表示计算机的源代码。

可以导致段落断开的标签（例如标题、<p> 和 <address> 标签）绝不能包含在 <pre> 所定义的块里。尽管有些浏览器会把段落结束标签解释为简单地换行，但是这种行为在所有浏览器上并不都是一样的。

pre 元素中允许的文本可以包括物理样式和基于内容的样式变化，还有链接、图像和水平分隔线。当把其他标签（比如 <a> 标签）放到 <pre> 块中时，就像放在 HTML/XHTML 文档的其他部分中一样即可。请看下面的例子：

```
<pre>
&lt;html&gt;

&lt;head&gt;
  &lt;script type=&quot;text/javascript&quot; src=&quot;loadxmldoc.js&quot;&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

  &lt;script type=&quot;text/javascript&quot;&gt;
    xmlDoc=<a href="dom_loadxmldoc.asp">loadXMLDoc</a>(&quot;books.xml&quot;);
    document.write(&quot;xmlDoc is loaded, ready for use&quot;);
  &lt;/script&gt;

&lt;/body&gt;

&lt;/html&gt;
</pre>
```

在上面的代码中，<pre> 标签中的特殊符号被转换为符号实体，比如 "<" 代表 "<"，">" 代表 ">"。另外，请注意蓝色的代码，我们在 <pre> 标签中使用了链接，也就是 <a> 标签。上面这段代码的显示效果如下：

```
<html>

<head>
  <script type="text/javascript" src="loadxmldoc.js">
</script>
</head>

<body>

  <script type="text/javascript">
    xmlDoc=[loadXMLDoc](/xmldom/dom_loadxmldoc.asp)("books.xml");
    document.write("xmlDoc is loaded, ready for use");
  </script>

</body>

</html>
```

提示和注释

提示：制表符（tab）在 <pre> 标签定义的块当中可以起到应有的作用，每个制表符占据 8 个字符的位置。但是我们不推荐使用它，因为在不同的浏览器中，Tab 的实现各不相同。在用 <pre> 标签格式化的文档段中使用空格，可以确保文本正确的水平位置。

提示：如果您希望使用 <pre> 标签来定义计算机源代码，比如 HTML 源代码，请使用符号实体来表示特殊字符，比如 "<" 代表 "<"，">" 代表 ">"，"&" 代表 "&"。

提示：在 W3School 中，非常多页面中的源代码实例都是通过 <pre> 标签定义的，您可以参考这些页面，学习如何使用该标签。我们甚至把 <pre> 标签与 <code> 标签结合起来使用，以获得更加精确的语义。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，pre 元素的 "width" 属性是不被赞成使用的。

在 XHTML 1.0 Strict DTD 中，pre 元素的 "width" 属性是不被支持的。

可选的属性

属性	值	描述
width	number	定义每行的最大字符数（通常是 40、80 或 132）。

全局属性

<pre> 标签支持 HTML 中的全局属性。

事件属性

<pre> 标签支持 HTML 中的事件属性。

TIY 实例

预格式文本

此例演示如何使用pre标签对空行和空格进行控制。

```
<html>

<body>

<pre>
这是
预格式文本。
它保留了      空格
和换行。
</pre>

<p>pre 标签很适合显示计算机代码：</p>

<pre>
for i = 1 to 10
     print i
next i
</pre>

</body>
</html>
```

HTML <progress> 标签

实例

正在进行的下载：

```
<progress value="22" max="100"></progress>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10, Firefox, Opera, Chrome 以及 Safari 6 支持 <progress> 标签。

注释：Internet Explorer 9 以及更早的版本不支持 <progress> 标签。

定义和用法

<progress> 标签标示任务的进度（进程）。

HTML 4.01 与 HTML 5 之间的差异

<progress> 标签是 HTML 5 中的新标签。

提示和注释

提示：请结合 <progress> 标签与 JavaScript 一同使用，来显示任务的进度。

注释：<progress> 标签不适合用来表示度量衡（例如，磁盘空间使用情况或查询结果）。如需表示度量衡，请使用 <meter> 标签代替。

属性

new : HTML5 中的新属性。

属性	值	描述
max	*number*	规定任务一共需要多少工作。
value	*number*	规定已经完成多少任务。

全局属性

<progress> 标签支持 HTML 中的全局属性。

事件属性

<progress> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Progress 对象

HTML <q> 标签

实例

标记短的引用：

```
<q>Here is a short quotation here is a short quotation</q>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <q> 标签。

定义和用法

<q> 标签定义短的引用。

浏览器经常在引用的内容周围添加引号。

<q> 与 <blockquote> 的区别

<q> 标签在本质上与 <blockquote> 是一样的。不同之处在于它们的显示和应用。<q> 标签用于简短的行内引用。如果需要从周围内容分离出来比较长的部分（通常显示为缩进的块），请使用 <blockquote> 标签。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

提示：根据 HTML 4.01 规范，q 元素应当使用分界引号来呈现，就是说，q 元素包含的文本必须以引号来开始和结束。Mozilla 浏览器 (Firefox) 以及 Opera 符合这个规定，但是 Internet Explorer 却不支持此规定。结果，如果要使用 <q> 标签，而且用自己的引号来满足 Internet

Explorer，那么就要在符合标准的浏览器使用两组引号。尽管如此，我们还是推荐使用 <q> 标签，这不仅仅因为我们喜欢标准，还因为我们预见到了将其应用于文档处理、信息提取等方面的显示效果。

可选的属性

属性	值	描述
cite	citation	定义引用的出处或来源（citation）

全局属性

<abbr> 标签支持 HTML 中的全局属性。

事件属性

<abbr> 标签支持 HTML 中的事件属性。

TIY 实例

引用

此例演示如何实现长短不一的引用语。

```
<html>

<body>

这是长的引用：
<blockquote>
这是长的引用。这是长的引用。这是长的引用。这是长的引用。这是长的引用。这是长的引用。这是长的引用。这是长的引
</blockquote>

这是短的引用：
<q>
这是短的引用。
</q>

<p>
使用 blockquote 元素的话，浏览器会插入换行和外边距，而 q 元素不会有任何特殊的呈现。
</p>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Quote 对象

HTML <rp> 标签

实例

一个 ruby 注释：

```
<ruby>
漢 <rt><rp>(</rp>ㄏㄢˋ<rp>)</rp></rt>
</ruby>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <rp> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <rp> 标签。

定义和用法

<rp> 标签在 ruby 注释中使用，以定义不支持 ruby 元素的浏览器所显示的内容。

ruby 注释是中文注音或字符。

在东亚使用，显示的是东亚字符的发音。

与 <ruby> 以及 <rt> 标签一同使用：

ruby 元素由一个或多个字符（需要一个解释/发音）和一个提供该信息的 rt 元素组成，还包括可选的 rp 元素，定义当浏览器不支持 "ruby" 元素时显示的内容。

HTML 4.01 与 HTML 5 之间的差异

<rp> 标签是 HTML 5 的新标签。

提示和注释

提示：支持 "ruby" 元素的浏览器不会显示 "rp" 元素的内容。

全局属性

<rp> 标签支持 HTML 中的全局属性。

事件属性

<rp> 标签支持 HTML 中的事件属性。

HTML <rt> 标签

实例

一个 ruby 注释：

```
<ruby>
漢 <rt> ㄏㄢˋ </rt>
</ruby>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <rt> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <rt> 标签。

定义和用法

<rt> 标签定义字符（中文注音或字符）的解释或发音。

ruby 注释是中文注音或字符。

在东亚使用，显示的是东亚字符的发音。

与 <ruby> 以及 <rt> 标签一同使用：

ruby 元素由一个或多个字符（需要一个解释/发音）和一个提供该信息的 rt 元素组成，还包括可选的 rp 元素，定义当浏览器不支持 "ruby" 元素时显示的内容。

HTML 4.01 与 HTML 5 之间的差异

<rt> 标签是 HTML 5 的新标签。

提示和注释

提示：支持 "ruby" 元素的浏览器不会显示 "rp" 元素的内容。

全局属性

<rt> 标签支持 HTML 中的全局属性。

事件属性

<rt> 标签支持 HTML 中的事件属性。

HTML <ruby> 标签

实例

一个 ruby 注释：

```
<ruby>
漢 <rt><rp>(</rp>ㄏㄢˋ<rp>)</rp></rt>
</ruby>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <ruby> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <ruby> 标签。

定义和用法

<ruby> 标签定义 ruby 注释（中文注音或字符）。

在东亚使用，显示的是东亚字符的发音。

与 <ruby> 以及 <rt> 标签一同使用：

ruby 元素由一个或多个字符（需要一个解释/发音）和一个提供该信息的 rt 元素组成，还包括可选的 rp 元素，定义当浏览器不支持 "ruby" 元素时显示的内容。

HTML 4.01 与 HTML 5 之间的差异

<ruby> 标签是 HTML 5 的新标签。

提示和注释

提示：支持 "ruby" 元素的浏览器不会显示 "rp" 元素的内容。

全局属性

<ruby> 标签支持 HTML 中的全局属性。

事件属性

<ruby> 标签支持 HTML 中的事件属性。

HTML <s> 标签

例子

可以像这样标记删除线文本：

```
在 HTML 5 中，<s>仍然支持</s>已经不支持这个标签了。
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <s> 标签。

定义和用法

<s> 标签可定义加删除线文本定义。

<s> 标签是 <strike> 标签的缩写版本，Netscape 和 Internet Explorer 都支持它。但在 HTML 4 和 XHTML 中已经不再赞成使用它了，意思就是不再使用了；它早晚有一天将会消失。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，<s> 标签不被赞成使用。

在 XHTML 1.0 Strict DTD 中，<s> 标签不被支持。

提示和注释：

注释：请使用 <del> 替代它！

全局属性

<s> 标签支持 HTML 中的全局属性。

事件属性

<s> 标签支持 HTML 中的事件属性。

TIY 实例

删除线文本

此例演示如何为文本标记添加删除线。

```
<html>

<body>

<p>在 HTML 5 中，<s>仍然支持</s>已经不支持这个标签了。</p>

</body>
</html>
```

参阅

请参阅 HTML 5 参考手册，了解有关 <s> 标签的最新信息：

HTML 5 <s> 标签

HTML <script> 标签

实例

在 HTML 页面中插入一段 JavaScript：

```
<script type="text/javascript">
document.write("Hello World!")
</script>
```

（在本页底部可以找到更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <script> 标签。

定义和用法

<script> 标签用于定义客户端脚本，比如 JavaScript。

script 元素既可以包含脚本语句，也可以通过 src 属性指向外部脚本文件。

必需的 type 属性规定脚本的 MIME 类型。

JavaScript 的常见应用时图像操作、表单验证以及动态内容更新。

提示和注释：

注释：假如此元素内部的代码没有位于某个函数中，那么这些代码会在页面被加载时被立即执行。<frameset> 标签之后的脚本会被忽略。

提示：请参阅 noscript 元素；对于那些在浏览器中禁用脚本或者其浏览器不支持客户端脚本的用户来说，该元素非常有用。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，script 元素的 "language" 属性不被赞成使用。

在 XHTML 1.0 Strict DTD 中，script 元素的 "language" 属性不被支持。

HTML 4 和 XHTML 在处理脚本中的内容方面有所不同：

在 HTML 4 中，内容类型声明为 CDATA，就是说不会对实体进行解析。

在 XHTML 中，内容类型声明为 (#PCDATA)，也就是说会对实体进行解析。

这意味着，在 XHTML 中，应该编码所有特殊的字符，或者把所有内容嵌套在 CDATA 部分中。

为了确保在 XHTML 文档中脚本正确进行解析，请使用如下语法：

```
<script type="text/javascript"><![CDATA[
  document.write("Hello World!")
//]]></script>
```

必选的属性

属性	值	描述
type	*MIME-type*	指示脚本的 MIME 类型。

可选的属性

属性	值	描述
async	async	规定异步执行脚本（仅适用于外部脚本）。
charset	*charset*	规定在外部脚本文件中使用的字符编码。
defer	defer	规定是否对脚本执行进行延迟，直到页面加载为止。
language	*script*	不赞成使用。规定脚本语言。请使用 type 属性代替它。
src	*URL*	规定外部脚本文件的 URL。
xml:space	preserve	规定是否保留代码中的空白。

全局属性

<script> 标签支持 HTML 中的全局属性。

TIY 实例

插入一段脚本

本例演示如何将脚本插入 HTML 文档。

```
<html>

<body>

<script type="text/javascript">
document.write("<h1>Hello World!</h1>")
</script>

</body>

</html>
```

运行于不支持脚本的浏览器

本例演示如何对付不支持脚本的浏览器。

```
<!DOCTYPE html>
<html>
<body>

<script type="text/javascript">
document.write("Hello World!")
</script>
<noscript>Sorry, your browser does not support JavaScript!</noscript>

<p>不支持 JavaScript 的浏览器将显示 noscript 元素中的文本。</p>

</body>
</html>
```

相关页面

教程：JavaScript 教程

教程：VBScript 教程

相关页面

HTML DOM 参考手册：Script 对象

HTML <section> 标签

实例

文档中的区段，解释了 PRC：

```
<section>
  <h1>PRC</h1>
  <p>The People's Republic of China was born in 1949...</p>
</section>
```

定义和用法

<section> 标签定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。

HTML 4.01 与 HTML 5 之间的差异

<section> 标签是 HTML 5 中的新标签。

属性

属性	值	描述
cite	*URL*	section 的 URL，假如 section 摘自 web 的话。

全局属性

<section> 标签支持 HTML 中的全局属性。

事件属性

<section> 标签支持 HTML 中的事件属性。

HTML <select> 标签

实例

创建带有 4 个选项的选择列表：

```
<select>
  <option value ="volvo">Volvo</option>
  <option value ="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <select> 标签。

定义和用法

select 元素可创建单选或多选菜单。

<select&> 元素中的 <option> 标签用于定义列表中的可用选项。

HTML 4.01 与 HTML 5 之间的差异

HTML5 添加了一些新属性。

提示和注释：

提示：select 元素是一种表单控件，可用于在表单中接受用户输入。

属性

New: HTML5 中的新属性。

属性	值	描述
autofocus	autofocus	规定在页面加载后文本区域自动获得焦点。
disabled	disabled	规定禁用该下拉列表。
form	*form_id*	规定文本区域所属的一个或多个表单。
multiple	multiple	规定可选择多个选项。
name	*name*	规定下拉列表的名称。
required	required	规定文本区域是必填的。
size	*number*	规定下拉列表中可见选项的数目。

全局属性

<select> 标签支持 HTML 中的全局属性。

事件属性

<select> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Select 对象

HTML <small> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <small> 标签。

定义和用法

<small> 标签呈现小号字体效果。

<small> 标签和它所对应的 <big> 标签一样，但它是缩小字体而不是放大。如果被包围的字体已经是字体模型所支持的最小字号，那么 <small> 标签将不起任何作用。

与 <big> 标签类似，<small> 标签也可以嵌套，从而连续地把文字缩小。每个 <small> 标签都把文本的字体变小一号，直到达到下限的一号字。

参阅

请参阅 HTML 5 参考手册，了解有关 <small> 标签的最新信息：

HTML 5 <small> 标签

HTML <source> 标签

实例

拥有两份源文件的音频播放器。浏览器应该选择它所支持的文件（如果有的话）：

```
<audio controls>
   <source src="horse.ogg" type="audio/ogg">
   <source src="horse.mp3" type="audio/mpeg">
 Your browser does not support the audio element.
</audio>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <source> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <source> 标签。

定义和用法

<source> 标签为媒介元素（比如 <video> 和 <audio>）定义媒介资源。

<source> 标签允许您规定可替换的视频/音频文件供浏览器根据它对媒体类型或者编解码器的支持进行选择。

HTML 4.01 与 HTML 5 之间的差异

<source> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
media	*media query*	规定媒体资源的类型。
src	*url*	规定媒体文件的 URL。
type	*numeric value*	规定媒体资源的 MIME 类型。

全局属性

<source> 标签支持 HTML 中的全局属性。

事件属性

<source> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Source 对象

HTML <span> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <span> 标签。

定义和用法

<span> 标签被用来组合文档中的行内元素。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

提示：请使用 <span> 来组合行内元素，以便通过样式来格式化它们。

注释：span 没有固定的格式表现。当对它应用样式时，它才会产生视觉上的变化。

例子

```
<p><span>some text.</span>some other text.</p>
```

例子解释

如果不对 span 应用样式，那么 span 元素中的文本与其他文本不会任何视觉上的差异。尽管如此，上例中的 span 元素仍然为 p 元素增加了额外的结构。

可以为 span 应用 id 或 class 属性，这样既可以增加适当的语义，又便于对 span 应用样式。

可以对同一个 <span> 元素应用 class 或 id 属性，但是更常见的情况是只应用其中一种。这两者的主要差异是，class 用于元素组（类似的元素，或者可以理解为某一类元素），而 id 用于标识单独的唯一的元素。

提示：事实上，您也许已经注意到了，W3School 站点上有一些文本的样式与其他文本是不同的。比如“提示”使用了粗体的橘红色。尽管实现这种效果的方法非常多，但是我们的做法是：使用“提示”使用 span 元素，然后对这个 span 元素的父元素，即 p 元素应用 class，这样就可以对这个类的子元素 span 应用相应的样式了。

HTML:

```
<p class="tip"><span>提示：</span>... ... ...</p>
```

CSS:

```
p.tip span {
    font-weight:bold;
    color:#ff9955;
    }
```

全局属性

<span> 标签支持 HTML 中的全局属性。

事件属性

<span> 标签支持 HTML 中的事件属性。

HTML <strike> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <strike> 标签。

定义和用法

<strike> 标签可定义加删除线文本定义。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，<strike> 标签不被赞成使用。

在 XHTML 1.0 Strict DTD 中，<strike> 标签不被支持。

提示和注释：

注释：请使用 <del> 替代它！

标准属性

```
id, class, title, style, dir, lang, xml:lang
```

如需完整的描述，请访问标准属性。

事件属性

```
onclick, ondblclick, onmousedown, onmouseup, onmouseover,
onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

如需完整的描述，请访问事件属性。

TIY 实例

删除线文本

此例演示如何为文本标记添加删除线。

```
<html>

<body>

<p>在 HTML 5 中，<strike>仍然支持</strike>已经不支持这个标签了。</p>

</body>
</html>
```

参阅

请参阅 HTML 5 参考手册，了解有关 <strike> 标签的最新信息：

HTML 5 <strike> 标签

HTML <style> 标签

实例

```
<html>
<head>
<style type="text/css">
h1 {color:red}
p {color:blue}
</style>
</head>

<body>
<h1>Header 1</h1>
<p>A paragraph.</p>
</body>
</html>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <style> 标签。

定义和用法

<style> 标签用于为 HTML 文档定义样式信息。

在 style 中，您可以规定在浏览器中如何呈现 HTML 文档。

type 属性是必需的，定义 style 元素的内容。唯一可能的值是 "text/css"。

style 元素位于 head 部分中。

提示和注释

提示：如需链接外部样式表，请使用 <link> 标签。

提示：如需学习更多有关样式表的知识，请阅读我们的 CSS 教程。

HTML 与 XHTML 之间的差异

NONE

必需的属性

属性	值	描述
type	text/css	规定样式表的 MIME 类型。

可选的属性

属性	值	描述
media	`screen` `tty` `tv` `projection` `handheld` `print` `braille` `aural` `all`	为样式表规定不同的媒介类型。

全局属性

<style> 标签支持 HTML 中的全局属性。

事件属性

<style> 标签支持 HTML 中的事件属性。

TIY 实例

HTML中的样式

本例演示如何使用添加到 <head> 部分的样式信息对 HTML 进行格式化。

```
<html>

<head>
<style type="text/css">
h1 {color: red}
p {color: blue}
</style>
</head>

<body>
<h1>header 1</h1>
<p>A paragraph.</p>
</body>

</html>
```

没有下划线的链接

本例演示如何使用样式属性做一个没有下划线的链接。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<meta http-equiv="Content-Language" content="zh-cn" />
</head>

<body>

<a href="/example/html/lastpage.html" style="text-decoration:none">
这是一个链接！
</a>

</body>
</html>
```

链接到一个外部样式表

本例演示如何<link>标签链接到一个外部样式表。

```
<html>

<head>
<link rel="stylesheet" type="text/css" href="/html/csstest1.css" >
</head>

<body>
<h1>我通过外部样式表进行格式化。</h1>
<p>我也一样！</p>
</body>

</html>
```

相关页面

HTML DOM 参考手册：Style 对象

HTML <sub> 标签

例子

```
这段文本包含 <sub>下标</sub>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <sub> 标签。

定义和用法

<sub> 标签可定义下标文本。

包含在 _{标签和其结束标签} 中的内容将会以当前文本流中字符高度的一半来显示，但是与当前文本流中文字的字体和字号都是一样的。

提示：无论是 <sub> 标签还是和它对应的 <sup> 标签，在数学等式、科学符号和化学公式中都非常有用。

HTML 与 XHTML 之间的差异

NONE

全局属性

<sub> 标签支持 HTML 中的全局属性。

事件属性

<sub> 标签支持 HTML 中的事件属性。

TIY 实例

上标和下标

本例演示如何使用 sup 和 sub 元素。

```
<html>
<body>

<p>
This text contains <sub>subscript</sub>
</p>

<p>
This text contains <sup>superscript</sup>
</p>

</body>
</html>
```

HTML <summary> 标签

实例

有关文档的详细信息：

```
<details>
<summary>HTML 5</summary>
This document teaches you everything you have to learn about HTML 5.
</details>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

只有 Chrome 以及 Safari 6 支持 <summary> 标签。

定义和用法

<summary> 标签包含 details 元素的标题，"details" 元素用于描述有关文档或文档片段的详细信息。

HTML 4.01 与 HTML 5 之间的差异

<summary> 标签是 HTML 5 中的新标签。

提示和注释

提示：请与 <details> 标签一起使用。标题是可见的，当用户点击标题时会显示出详细信息。

注释："summary" 元素应该是 "details" 元素的第一个子元素。

全局属性

<summary> 标签支持 HTML 中的全局属性。

事件属性

<summary> 标签支持 HTML 中的事件属性。

HTML <sup> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <sup> 标签。

定义和用法

<sup> 标签可定义上标文本。

包含在 ^{标签和其结束标签} 中的内容将会以当前文本流中字符高度的一半来显示，但是与当前文本流中文字的字体和字号都是一样的。

提示：这个标签在向文档添加脚注以及表示方程式中的指数值时非常有用。如果和 <a> 标签结合起来使用，就可以创建出很好的超链接脚注。

HTML 与 XHTML 之间的差异

NONE

例子

```
这段文本包含 <sup>上标</sup>
```

全局属性

<sup> 标签支持 HTML 中的全局属性。

事件属性

<sup> 标签支持 HTML 中的事件属性。

HTML <table> 标签

实例

一个简单的 HTML 表格，包含两行两列：

```
<table border="1">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <table> 标签。

定义和用法

<table> 标签定义 HTML 表格。

简单的 HTML 表格由 table 元素以及一个或多个 tr、th 或 td 元素组成。

tr 元素定义表格行，th 元素定义表头，td 元素定义表格单元。

更复杂的 HTML 表格也可能包括 caption、col、colgroup、thead、tfoot 以及 tbody 元素。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，table 元素的 "align" 和 "bgcolor" 属性是不被赞成使用的。

在 XHTML 1.0 Strict DTD，table 元素的 "align" 和 "bgcolor" 属性是不被支持的。

可选的属性

属性	值	描述
align	left center right	不赞成使用。请使用样式代替。规定表格相对周围元素的对齐方式。
bgcolor	*rgb(x,x,x) #xxxxxx colorname*	不赞成使用。请使用样式代替。规定表格的背景颜色。
border	*pixels*	规定表格边框的宽度。
cellpadding	*pixels %*	规定单元边沿与其内容之间的空白。
cellspacing	*pixels %*	规定单元格之间的空白。
frame	void above below hsides lhs rhs vsides box border	规定外侧边框的哪个部分是可见的。
rules	none groups rows cols all	规定内侧边框的哪个部分是可见的。
summary	*text*	规定表格的摘要。
width	*% pixels*	规定表格的宽度。

全局属性

<table> 标签支持 HTML 中的全局属性。

事件属性

<table> 标签支持 HTML 中的事件属性。

TIY 实例

表格

这个例子演示如何在 HTML 文档中创建表格。

```
<html>

<body>

<p>每个表格由 table 标签开始。</p>
<p>每个表格行由 tr 标签开始。</p>
<p>每个表格数据由 td 标签开始。</p>

<h4>一列：</h4>
<table border="1">
<tr>
  <td>100</td>
</tr>
</table>

<h4>一行三列：</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
</table>

<h4>两行三列：</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>

</body>
</html>
```

表格边框

本例演示各种类型的表格边框。

```
<html>

<body>

<h4>带有普通的边框：</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>带有粗的边框：</h4>
<table border="8">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>带有很粗的边框：</h4>
<table border="15">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

</body>
</html>
```

没有边框的表格

本例演示一个没有边框的表格。

```
<html>

<body>

<h4>这个表格没有边框：</h4>
<table>
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>

<h4>这个表格也没有边框：</h4>
<table border="0">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>

</body>
</html>
```

表格中的标题(Headings)

本例演示如何显示表格标题。

```
<html>

<body>

<h4>表头：</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <th>电话</th>
  <th>电话</th>
</tr>
<tr>
  <td>Bill Gates</td>
  <td>555 77 854</td>
  <td>555 77 855</td>
</tr>
</table>

<h4>垂直的表头：</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <td>Bill Gates</td>
</tr>
<tr>
  <th>电话</th>
  <td>555 77 854</td>
</tr>
<tr>
  <th>电话</th>
  <td>555 77 855</td>
</tr>
</table>

</body>
</html>
```

空单元格

本例展示如何使用 " " 处理没有内容的单元格。

```
<html>

<body>

<table border="1">
<tr>
  <td>Some text</td>
  <td>Some text</td>
</tr>
<tr>
  <td></td>
  <td>Some text</td>
</tr>
</table>

<p>正如您看到的，其中一个单元没有边框。这是因为它是空的。在该单元中插入一个空格后，仍然没有边框。</p>

<p>我们的技巧是在单元中插入一个 no-breaking 空格。</p>

<p>no-breaking 空格是一个字符实体。如果您不清楚什么是字符实体，请阅读关于字符实体的章节。</p>

<p>no-breaking 空格由和号开始 ("&")，然后是字符"nbsp"，并以分号结尾(";")。</p>

</body>
</html>
```

带有标题的表格

本例演示一个带标题 (caption) 的表格

```
<html>

<body>

<h4>这个表格有一个标题，以及粗边框：</h4>

<table border="6">
<caption>我的标题</caption>
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>

</body>
```

跨行或跨列的表格单元格

本例演示如何定义跨行或跨列的表格单元格。

```
<html>

<body>

<h4>横跨两列的单元格：</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <th colspan="2">电话</th>
</tr>
<tr>
  <td>Bill Gates</td>
  <td>555 77 854</td>
  <td>555 77 855</td>
</tr>
</table>

<h4>横跨两行的单元格：</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <td>Bill Gates</td>
</tr>
<tr>
  <th rowspan="2">电话</th>
  <td>555 77 854</td>
</tr>
<tr>
  <td>555 77 855</td>
</tr>
</table>

</body>
</html>
```

表格内的标签

本例演示如何在不同的元素内显示元素。

```
<html>

<body>

<table border="1">
<tr>
  <td>
   <p>这是一个段落。</p>
   <p>这是另一个段落。</p>
  </td>
  <td>这个单元包含一个表格：
   <table border="1">
   <tr>
     <td>A</td>
     <td>B</td>
   </tr>
   <tr>
     <td>C</td>
     <td>D</td>
   </tr>
   </table>
  </td>
</tr>
<tr>
  <td>这个单元包含一个列表：
   <ul>
    <li>苹果</li>
    <li>香蕉</li>
    <li>菠萝</li>
   </ul>
  </td>
  <td>HELLO</td>
</tr>
</table>

</body>
</html>
```

单元格填充(Cell padding)

本例演示如何使用单元格填充来创建单元格内容与其边框之间的空白。

```
<html>

<body>

<h4>没有 cellpadding：</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>带有 cellpadding：</h4>
<table border="1"
cellpadding="10">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

</body>
</html>
```

单元格间距(Cell spacing)

本例演示如何使用单元格间距增加单元格之间的距离。

```
<html>

<body>

<h4>没有 cellspacing：</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>带有 cellspacing：</h4>
<table border="1"
cellspacing="10">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

</body>
</html>
```

向表格添加背景颜色或背景图像

本例演示如何向表格添加背景

```
<html>

<body>

<h4>背景颜色：</h4>
<table border="1"
bgcolor="red">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>背景图像：</h4>
<table border="1"
background="/i/eg_bg_07.gif">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

</body>
</html>
```

向表格单元添加背景颜色或者背景图像

本例演示如何向一个或者更多表格单元添加背景

```
#### 单元背景：

   First |
   Row |

  Second |
   Row |
```

在表格单元中排列内容

本例演示如何使用 "align" 属性排列单元格内容,以便创建一个美观的表格。

```
<html>

<body>

<table width="400" border="1">
 <tr>
  <th align="left">消费项目....</th>
  <th align="right">一月</th>
  <th align="right">二月</th>
 </tr>
 <tr>
  <td align="left">衣服</td>
  <td align="right">$241.10</td>
  <td align="right">$50.20</td>
 </tr>
 <tr>
  <td align="left">化妆品</td>
  <td align="right">$30.00</td>
  <td align="right">$44.45</td>
 </tr>
 <tr>
  <td align="left">食物</td>
  <td align="right">$730.40</td>
  <td align="right">$650.00</td>
 </tr>
 <tr>
  <th align="left">总计</th>
  <th align="right">$1001.50</th>
  <th align="right">$744.65</th>
 </tr>
</table>

</body>
</html>
```

框架(frame)属性

本例演示如何使用 "frame" 属性来控制围绕表格的边框。

```
<html>
<body>

<p><b>注释：</b>frame 属性无法在 Internet Explorer 中正确地显示。</p>

<p>Table with frame="box":</p>
<table frame="box">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>

<p>Table with frame="above":</p>
<table frame="above">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>

<p>Table with frame="below":</p>
<table frame="below">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>

<p>Table with frame="hsides":</p>
<table frame="hsides">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>

<p>Table with frame="vsides":</p>
<table frame="vsides">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>

</body>
</html>
```

相关页面

HTML DOM 参考手册：Table 对象

HTML <tbody> 标签

实例

带有 thead、tbody 以及 tfoot 元素的 HTML 表格：

```
<table border="1">
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>

  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>

  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

<tbody> 标签仅得到所有主流浏览器的部分支持。

定义和用法

<tbody> 标签表格主体（正文）。该标签用于组合 HTML 表格的主体内容。

tbody 元素应该与 thead 和 tfoot 元素结合起来使用。

thead 元素用于对 HTML 表格中的表头内容进行分组，而 tfoot 元素用于对 HTML 表格中的表注（页脚）内容进行分组。

注释：如果您使用 thead、tfoot 以及 tbody 元素，您就必须使用全部的元素。它们的出现次序是：thead、tfoot、tbody，这样浏览器就可以在收到所有数据前呈现页脚了。您必须在 table 元素内部使用这些标签。

提示：在默认情况下这些元素不会影响到表格的布局。不过，您可以使用 CSS 使这些元素改变表格的外观。

详细描述

thead、tfoot 以及 tbody 元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：<thead> 内部必须拥有 <tr> 标签！

注释：<thead>、<tbody> 以及 <tfoot> 很少被使用，这是因为糟糕的浏览器支持。我们期望在 XHTML 的未来版本中这种情况会发生变化。假如您使用 Internet Explorer 5.0 或更新的浏览器，可以在我们的 XML 教程中查看一个例子。

可选的属性

属性	值	描述
align	`right` `left` `center` `justify` `char`	定义 tbody 元素中内容的对齐方式。
char	*character*	规定根据哪个字符来进行文本对齐。
charoff	*number*	规定第一个对齐字符的偏移量。
valign	`top` `middle` `bottom` `baseline`	规定 tbody 元素中内容的垂直对齐方式。

全局属性

<tbody> 标签支持 HTML 中的全局属性。

事件属性

<tbody> 标签支持 HTML 中的事件属性。

HTML <td> 标签

实例

一个简单的 HTML 表格，包含两行两列：

```
<table border="1">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <td> 标签。

定义和用法

<td> 标签定义 HTML 表格中的标准单元格。

HTML 表格有两类单元格：

- 表头单元 - 包含头部信息（由 th 元素创建）
- 标准单元 - 包含数据（由 td 元素创建）

td 元素中的文本一般显示为正常字体且左对齐。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，td 元素的 "bgcolor"、"height"、"width" 以及 "nowrap" 是不被赞成使用的。

在 XHTML 1.0 Strict DTD 中，td 元素的 "bgcolor"、"height"、"width" 以及 "nowrap" 是不被支持的。

提示和注释

提示：请使用 colspan 和 rowspan 属性来实现内容横跨多个行或列。

可选的属性

属性	值	描述
abbr	*text*	规定单元格中内容的缩写版本。
align	`left` `right` `center` `justify` `char`	规定单元格内容的水平对齐方式。
axis	*category_name*	对单元进行分类。
bgcolor	*rgb(x,x,x) #xxxxxx colorname*	不赞成使用。请使用样式取而代之。规定单元格的背景颜色。
char	*character*	规定根据哪个字符来进行内容的对齐。
charoff	*number*	规定对齐字符的偏移量。
colspan	*number*	规定单元格可横跨的列数。
headers	*headercells'id*	规定与单元格相关的表头。
height	*pixels %*	不赞成使用。请使用样式取而代之。规定表格单元格的高度。
nowrap	nowrap	不赞成使用。请使用样式取而代之。规定单元格中的内容是否折行。
rowspan	*number*	规定单元格可横跨的行数。
scope	`col` `colgroup` `row` `rowgroup`	定义将表头数据与单元数据相关联的方法。
valign	`top` `middle` `bottom` `baseline`	规定单元格内容的垂直排列方式。
width	*pixels %*	不赞成使用。请使用样式取而代之。规定表格单元格的宽度。

全局属性

<td> 标签支持 HTML 中的全局属性。

事件属性

<td> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：TableData 对象

HTML <textarea> 标签

实例

```
<textarea rows="3" cols="20">
在w3school，你可以找到你所需要的所有的网站建设教程。
</textarea>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <textarea> 标签。

定义和用法

<textarea> 标签定义多行的文本输入控件。

文本区中可容纳无限数量的文本，其中的文本的默认字体是等宽字体（通常是 Courier）。

可以通过 cols 和 rows 属性来规定 textarea 的尺寸，不过更好的办法是使用 CSS 的 height 和 width 属性。

注释：在文本输入区内的文本行间，用 "%OD%OA" （回车/换行）进行分隔。

提示：可以通过 <textarea> 标签的 wrap 属性设置文本输入区内的换行模式。有关 wrap 属性的详细信息。

HTML 与 XHTML 之间的差异

NONE

属性

New: HTML5 中的新属性。

属性	值	描述
autofocus	autofocus	规定在页面加载后文本区域自动获得焦点。
cols	*number*	规定文本区内的可见宽度。
disabled	disabled	规定禁用该文本区。
form	*form_id*	规定文本区域所属的一个或多个表单。
maxlength	*number*	规定文本区域的最大字符数。
name	*name_of_textarea*	规定文本区的名称。
placeholder	*text*	规定描述文本区域预期值的简短提示。
readonly	readonly	规定文本区为只读。
required	required	规定文本区域是必填的。
rows	*number*	规定文本区内的可见行数。
wrap	`hard` `soft`	规定当在表单中提交时，文本区域中的文本如何换行。。

全局属性

<textarea> 标签支持 HTML 中的全局属性。

事件属性

<textarea> 标签支持 HTML 中的事件属性。

HTML <tfoot> 标签

实例

带有 thead、tbody 以及 tfoot 元素的 HTML 表格：

```
<table border="1">
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>

  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>

  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

<tfoot> 标签仅得到所有主流浏览器的部分支持。

定义和用法

<tfoot> 标签定义表格的页脚（脚注或表注）。该标签用于组合 HTML 表格中的表注内容。

tfoot 元素应该与 thead 和 tbody 元素结合起来使用。

thead 元素用于对 HTML 表格中的表头内容进行分组，而 tbody 元素用于对 HTML 表格中的主体内容进行分组。

注释：如果您使用 thead、tfoot 以及 tbody 元素，您就必须使用全部的元素。它们的出现次序是：thead、tfoot、tbody，这样浏览器就可以在收到所有数据前呈现页脚了。您必须在 table 元素内部使用这些标签。

提示：在默认情况下这些元素不会影响到表格的布局。不过，您可以使用 CSS 使这些元素改变表格的外观。

详细描述

thead、tfoot 以及 tbody 元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：<thead> 内部必须拥有 <tr> 标签！

注释：<thead>、<tbody> 以及 <tfoot> 很少被使用，这是因为糟糕的浏览器支持。我们期望在 XHTML 的未来版本中这种情况会发生变化。假如您使用 Internet Explorer 5.0 或更新的浏览器，可以在我们的 XML 教程中查看一个例子。

可选的属性

属性	值	描述
align	`right` `left` `center` `justify` `char`	定义 tfoot 元素中内容的对齐方式。
char	*character*	规定根据哪个字符来进行文本对齐。
charoff	*number*	规定第一个对齐字符的偏移量。
valign	`top` `middle` `bottom` `baseline`	规定 tfoot 元素中内容的垂直对齐方式。

全局属性

<tfoot> 标签支持 HTML 中的全局属性。

事件属性

<tfoot> 标签支持 HTML 中的事件属性。

HTML <th> 标签

实例

普通的 HTML 表格，包含两行两列：

```
<table border="1">
  <tr>
    <th>Company</th>
    <th>Address</th>
  </tr>

  <tr>
    <td>Apple, Inc.</td>
    <td>1 Infinite Loop Cupertino, CA 95014</td>
  </tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <th> 标签。

定义和用法

定义表格内的表头单元格。

HTML 表单中有两种类型的单元格：

- 表头单元格 - 包含表头信息（由 th 元素创建）
- 标准单元格 - 包含数据（由 td 元素创建）

th 元素内部的文本通常会呈现为居中的粗体文本，而 td 元素内的文本通常是左对齐的普通文本。

提示和注释

提示：如果需要将内容横跨多个行或列，请使用 colspan 和 rowspan 属性。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，th 元素的 "bgcolor"、"height"、"width" 以及 "nowrap" 属性是不被推荐使用的。

在 XHTML 1.0 Strict DTD 中，th 元素的 "bgcolor"、"height"、"width" 以及 "nowrap" 属性是不被支持。

可选的属性

属性	值	描述
abbr	*text*	规定单元格中内容的缩写版本。
align	`left` `right` `center` `justify` `char`	规定单元格内容的水平对齐方式。
axis	*category_name*	对单元格进行分类。
bgcolor	*rgb(x,x,x)* *#xxxxxx* *colorname*	不推荐使用。请使用样式替代它。规定表格单元格的背景颜色。
char	*character*	规定根据哪个字符来进行内容的对齐。
charoff	*number*	规定对齐字符的偏移量。
colspan	*number*	设置单元格可横跨的列数。
headers	*idrefs*	由空格分隔的表头单元格 ID 列表，为数据单元格提供表头信息。
height	*pixels* *%*	不推荐使用。请使用样式替代它。规定表格单元格的高度。
nowrap	nowrap	不推荐使用。请使用样式取而代之。规定单元格中的内容是否折行。
rowspan	*number*	规定单元格可横跨的行数。
scope	`col` `colgroup` `row` `rowgroup`	定义将表头数据与单元数据相关联的方法。
valign	`top` `middle` `bottom` `baseline`	规定单元格内容的垂直排列方式。
width	*pixels* *%*	不推荐使用。请使用样式取而代之。规定表格单元格的宽度。

全局属性

<th> 标签支持 HTML 中的全局属性。

事件属性

<th> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：TableHeader 对象

HTML <thead> 标签

实例

带有 thead、tbody 以及 tfoot 元素的 HTML 表格：

```
<table border="1">
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>

  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>

  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

<thead> 标签仅得到所有主流浏览器的部分支持。

定义和用法

<thead> 标签定义表格的表头。该标签用于组合 HTML 表格的表头内容。

thead 元素应该与 tbody 和 tfoot 元素结合起来使用。

tbody 元素用于对 HTML 表格中的主体内容进行分组，而 tfoot 元素用于对 HTML 表格中的表注（页脚）内容进行分组。

注释：如果您使用 thead、tfoot 以及 tbody 元素，您就必须使用全部的元素。它们的出现次序是：thead、tfoot、tbody，这样浏览器就可以在收到所有数据前呈现页脚了。您必须在 table 元素内部使用这些标签。

提示：在默认情况下这些元素不会影响到表格的布局。不过，您可以使用 CSS 使这些元素改变表格的外观。

详细描述

thead、tfoot 以及 tbody 元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

HTML 与 XHTML 之间的差异

NONE

提示和注释：

注释：<thead> 内部必须拥有 <tr> 标签！

注释：<thead>、<tbody> 以及 <tfoot> 很少被使用，这是因为糟糕的浏览器支持。我们期望在 XHTML 的未来版本中这种情况会发生变化。假如您使用 Internet Explorer 5.0 或更新的浏览器，可以在我们的 XML 教程中查看一个例子。

可选的属性

属性	值	描述
align	`right` `left` `center` `justify` `char`	定义 thead 元素中内容的对齐方式。
char	*character*	规定根据哪个字符来进行文本对齐。
charoff	*number*	规定第一个对齐字符的偏移量。
valign	`top` `middle` `bottom` `baseline`	规定 thead 元素中内容的垂直对齐方式。

全局属性

<thead> 标签支持 HTML 中的全局属性。

事件属性

<thead> 标签支持 HTML 中的事件属性。

HTML <time> 标签

实例

如何定义时间和日期：

```
<p>我们在每天早上 <time>9:00</time> 开始营业。</p>

<p>我在 <time datetime="2008-02-14">情人节</time> 有个约会。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

<time> 标签不会在任何浏览器中呈现任何特殊效果。

定义和用法

<time> 标签定义公历的时间（24 小时制）或日期，时间和时区偏移是可选的。

该元素能够以机器可读的方式对日期和时间进行编码，这样，举例说，用户代理能够把生日提醒或排定的事件添加到用户日程表中，搜索引擎也能够生成更智能的搜索结果。

浏览器支持

目前所有主流浏览器都不支持 <time> 标签。

HTML 4.01 与 HTML 5 之间的差异

<time> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
datetime	*datetime*	规定日期 / 时间。否则，由元素的内容给定日期 / 时间。
pubdate	*pubdate*	指示 <time> 元素中的日期 / 时间是文档（或 <article> 元素）的发布日期。

全局属性

<time> 标签支持 HTML 中的全局属性。

事件属性

<time> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Time 对象

HTML <title> 标签

实例

一个简单的 HTML 文档，带有尽可能少的必需的标签：

```
<html>

  <head>
    <title>XHTML Tag Reference</title>
  </head>

  <body>
    The content of the document......
  </body>

</html>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <title> 标签。

定义和用法

<title> 元素可定义文档的标题。

浏览器会以特殊的方式来使用标题，并且通常把它放置在浏览器窗口的标题栏或状态栏上。同样，当把文档加入用户的链接列表或者收藏夹或书签列表时，标题将成为该文档链接的默认名称。

HTML 与 XHTML 之前的差异

NONE

提示和注释

提示：<title> 标签是 <head> 标签中唯一要求包含的东西。

标准属性

<title> 标签支持以下标准属性：

属性	值	描述
dir	`rtl` `ltr`	规定元素中内容的文本方向。
lang	*language_code*	规定元素中内容的语言代码。
xml:lang	*language_code*	规定 XHTML 文档中元素内容的语言代码。

DTD：指示在哪种 XHTML 1.0 DTD 中允许该标签。S=Strict, T=Transitional, F=Frameset.

如需更多有关 HTML 标准属性的信息，请访问标准属性。

全局属性

<title> 标签支持 HTML 中的全局属性。

延伸阅读：标题里是什么？

一定要选择一个正确的标题，这对于定义文档并确保它能够在 Web 上有效利用来说是十分重要的。

请记住，用户可以用任何顺序、独立地访问文档集中的每一个文档。所以，文档的标题不仅应当根据其他文档的上下文定义，而且还要显示其自身的特点。

含有文档引用排序的标题通常不是什么好标题。举个例子，像“第十六章”或“第五部分”这样的标题，对读者理解其内容方面毫无用处。描述性更强的标题，像“第十六章：HTML 标题”，或者“第五部分：如何使用标题”，这样的标题不仅表达了它在一个大型文档集中的位置，还说明了文档的具体内容，吸引读者更有兴趣读下去。

自我引用的标题也没有什么用处。像“主页”这样的标题和内容毫无关系，类似的还有“反馈页”或“常用链接”等。你应该设计一个能够传达一定内容和目的的标题，令读者凭这个标题就可以判断是否由必要访问这个页面。“HTML <title> 标签的详细信息”，这就是一个描述性的标题，类似的还有“HTML <title> 标签的反馈页”等等。

人们常常会花费大量时间去创建 Web 文档，但却经常只是因为一个不吸引人或无意义的标题，而把这些努力全都浪费掉了。当自动为用户搜集链接的特殊软件在 Web 上越来越流行时，只有网页的标题才会作为与页面相关的描述性词语，被插入到庞大的链接数据库中。因此，我们怎么强调这一点都不过分：请为自己的每个文档都认真地选择一个描述性的、实用的并与上下文独立的标题。

相关页面

HTML DOM 参考手册：Title 对象

HTML <tr> 标签

实例

一个简单的 HTML 表格，包含两行两列：

```
<table border="1">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <tr> 标签。

定义和用法

<tr> 标签定义 HTML 表格中的行。

tr 元素包含一个或多个 th 或 td 元素。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，tr 元素的 "bgcolor" 是不被赞成使用的。

在 XHTML 1.0 Strict DTD 中，tr 元素的 "bgcolor" 是不被支持的。

可选的属性

属性	值	描述
align	`right` `left` `center` `justify` `char`	定义表格行的内容对齐方式。
bgcolor	*rgb(x,x,x) #xxxxxx colorname*	不赞成使用。请使用样式取而代之。规定表格行的背景颜色。
char	*character*	规定根据哪个字符来进行文本对齐。
charoff	*number*	规定第一个对齐字符的偏移量。
valign	`top` `middle` `bottom` `baseline`	规定表格行中内容的垂直对齐方式。

全局属性

<tr> 标签支持 HTML 中的全局属性。

事件属性

<tr> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：TableRow 对象

HTML <track> 标签

实例

播放带有字幕的视频：

```
<video width="320" height="240" controls="controls">
  <source src="forrest_gump.mp4" type="video/mp4" />
  <source src="forrest_gump.ogg" type="video/ogg" />
  <track kind="subtitles" src="subs_chi.srt" srclang="zh" label="Chinese">
  <track kind="subtitles" src="subs_eng.srt" srclang="en" label="English">
</video>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10, Chrome 以及 Opera 支持 <track> 标签。

定义和用法

<track> 标签为诸如 video 元素之类的媒介规定外部文本轨道。

用于规定字幕文件或其他包含文本的文件，当媒介播放时，这些文件是可见的。

浏览器支持

目前所有主流浏览器都不支持 <track> 标签。

HTML 4.01 与 HTML 5 之间的差异

<track> 标签是 HTML 5 中的新标签。

属性

new : HTML5 中的新属性。

属性	值	描述
default	default	规定该轨道是默认的，假如没有选择任何轨道。
kind	`captions` `chapters` `descriptions` `metadata` `subtitles`	表示轨道属于什么文本类型。
label	*label*	轨道的标签或标题。
src	*url*	轨道的 URL。
srclang	*language_code*	轨道的语言，若 kind 属性值是 "subtitles"，则该属性必需的。

全局属性

<track> 标签支持 HTML 中的全局属性。

事件属性

<track> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Track 对象

HTML <tt> 标签

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <tt> 标签。

定义和用法

<tt> 标签呈现类似打字机或者等宽的文本效果。

<tt> 标签与 <code> 和 <kbd> 标签一样，<tt> 标签和必需的 </tt> 结束标签告诉浏览器，要把其中包含的文本显示为等宽字体。对于那些已经使用了等宽字体的浏览器来说，这个标签在文本的显示上就没有什么特殊效果了。

参阅

请参阅 HTML 5 参考手册，了解有关 <tt> 标签的最新信息：

HTML 5 <tt> 标签

HTML <u> 标签

实例

使用 <u> 标签为文本添加下划线：

```
<p>如果文本不是超链接，就不要<u>对其使用下划线</u>。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <u> 标签。

定义和用法

<u> 标签可定义下划线文本。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，u 元素是不被推荐使用的。

在 XHTML 1.0 Strict DTD 中，u 元素是不被支持的。

提示和注释：

注释：请尽量避免为文本加下划线 - 用户会把它混淆为一个超链接！

全局属性

<u> 标签支持 HTML 中的全局属性。

事件属性

<u> 标签支持 HTML 中的事件属性。

HTML <ul> 标签

实例

无序 HTML 列表：

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

（请在页面底部查看更多实例）

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 <ul> 标签。

定义和用法

 标签定义无序列表。

HTML 与 XHTML 之间的差异

在 HTML 4.01 中，ul 元素的 "compact" 和 "type" 属性是不被赞成使用的。

在 XHTML 1.0 Strict DTD 中，ul 元素的 "compact" 和 "type" 属性是不被支持的。

提示和注释：

提示：请使用样式来定义列表的类型。

可选的属性

属性	值	描述
compact	compact	不赞成使用。请使用样式取代它。规定列表呈现的效果比正常情况更小巧。
type	`disc` `square` `circle`	不赞成使用。请使用样式取代它。规定列表的项目符号的类型。

全局属性

 标签支持 HTML 中的全局属性。

事件属性

 标签支持 HTML 中的事件属性。

HTML <video> 标签

实例

一段简单的 HTML5 视频：

```
<video src="movie.ogg" controls="controls">
您的浏览器不支持 video 标签。
</video>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome 以及 Safari 支持 <video> 标签。

注释：Internet Explorer 8 以及更早的版本不支持 <video> 标签。

定义和用法

<video> 标签定义视频，比如电影片段或其他视频流。

HTML 4.01 与 HTML 5 之间的差异

<video> 标签是 HTML 5 的新标签。

提示和注释

提示：可以在开始标签和结束标签之间放置文本内容，这样老的浏览器就可以显示出不支持该标签的信息。

属性

new : HTML5 中的新属性。

属性	值	描述
autoplay	autoplay	如果出现该属性，则视频在就绪后马上播放。
controls	controls	如果出现该属性，则向用户显示控件，比如播放按钮。
height	*pixels*	设置视频播放器的高度。
loop	loop	如果出现该属性，则当媒介文件完成播放后再次开始播放。
muted	muted	规定视频的音频输出应该被静音。
poster	*URL*	规定视频下载时显示的图像，或者在用户点击播放按钮前显示的图像。
preload	preload	如果出现该属性，则视频在页面加载时进行加载，并预备播放。如果使用 "autoplay"，则忽略该属性。
src	*url*	要播放的视频的 URL。
width	*pixels*	设置视频播放器的宽度。

全局属性

<video> 标签支持 HTML 中的全局属性。

事件属性

<video> 标签支持 HTML 中的事件属性。

相关页面

HTML DOM 参考手册：Video 对象

HTML <wbr> 标签

实例

一段带有 Word Break Opportunity 的文本：

```
<p>
如果想学习 AJAX，那么您必须熟悉 XML<wbr>Http<wbr>Request 对象。
</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有浏览器都支持 <wbr> 标签，除了 Internet Explorer。

定义和用法

Word Break Opportunity (<wbr>) 规定在文本中的何处适合添加换行符。

提示：如果单词太长，或者您担心浏览器会在错误的位置换行，那么您可以使用 <wbr> 元素来添加 Word Break Opportunity（单词换行时机）。

浏览器支持

所有主流浏览器均支持 <wbr> 标签，除了 Internet Explorer。

HTML 4.01 与 HTML 5 之间的差异

<wbr> 标签是 HTML 5 中的新标签。

全局属性

<wbr> 标签支持 HTML 中的全局属性。

事件属性

<wbr> 标签支持 HTML 中的事件属性。

HTML 全局属性

HTML 属性赋予元素意义和语境。

下面的全局属性可用于任何 HTML 元素。

HTML 全局属性

= HTML5 中添加的属性。

属性	描述
accesskey	规定激活元素的快捷键。
class	规定元素的一个或多个类名（引用样式表中的类）。
contenteditable	规定元素内容是否可编辑。
contextmenu	规定元素的上下文菜单。上下文菜单在用户点击元素时显示。
data-*	用于存储页面或应用程序的私有定制数据。
dir	规定元素中内容的文本方向。
draggable	规定元素是否可拖动。
dropzone	规定在拖动被拖动数据时是否进行复制、移动或链接。
hidden	规定元素仍未或不再相关。
id	规定元素的唯一 id。
lang	规定元素内容的语言。
spellcheck	规定是否对元素进行拼写和语法检查。
style	规定元素的行内 CSS 样式。
tabindex	规定元素的 tab 键次序。
title	规定有关元素的额外信息。
translate	规定是否应该翻译元素内容。

HTML accesskey 属性

HTML 全局属性

实例

带有指定快捷键的超链接：

```
<a href="http://www.w3school.com.cn/html/" accesskey="h">HTML</a><br />
<a href="http://www.w3school.com.cn/css/" accesskey="c">CSS</a>
```

浏览器支持

几乎所有浏览器均 accesskey 属性，除了 Opera。

定义和用法

accesskey 属性规定激活（使元素获得焦点）元素的快捷键。

提示和注释

注释：以下元素支持 accesskey 属性：<a>, <area>, <button>, <input>, <label>, <legend> 以及 <textarea>。

语法

```
<element accesskey="_character_">
```

属性值

值	描述
character	规定激活（使元素获得焦点）元素的便捷按键。

HTML 全局属性

HTML class 属性

HTML 全局属性

实例

在 HTML 文档中使用 class 属性：

```
<html>
<head>
<style type="text/css">
h1.intro {color:blue;}
p.important {color:green;}
</style>
</head>

<body>
<h1 class="intro">Header 1</h1>
<p>A paragraph.</p>
<p class="important">Note that this is an important paragraph.</p>
</body>
</html>
```

（可以在本页底部找到更多实例）

定义和用法

class 属性规定元素的类名（classname）。

class 属性大多数时候用于指向样式表中的类（class）。不过，也可以利用它通过 JavaScript 来改变带有指定 class 的 HTML 元素。

提示和注释

注释：class 属性不能在以下 HTML 元素中使用：base, head, html, meta, param, script, style 以及 title。

提示：可以给 HTML 元素赋予多个 class，例如：<span class="left_menu important">。这么做可以把若干个 CSS 类合并到一个 HTML 元素。

提示：类名不能以数字开头！只有 Internet Explorer 支持这种做法。

浏览器支持

W3C: "W3C" 列指示 W3C 的 HTML/XHTML 推荐标准中是否定义了该属性。

IE	Firefox	Opera	Safari	W3C
YES	YES	YES	YES	YES

语法

```
<element class="_value_">
```

属性值

值	描述
classname	规定元素的类的名称。如需为一个元素规定多个类，用空格分隔类名。

亲自试一试 - 实例

向元素添加多个类

如何向一个元素添加多个类。

HTML 全局属性

HTML contenteditable 属性

HTML 全局属性

实例

一段可编辑的段落：

```
<p contenteditable="true">这是一个可编辑的段落。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 contenteditable 属性。

定义和用法

contenteditable 属性规定元素内容是否可编辑。

注释：如果元素未设置 contenteditable 属性，那么元素会从其父元素继承该属性。

HTML 4.01 与 HTML5 之间的差异

contenteditable 属性是 HTML5 中的新属性。

语法

```
<_element_ contenteditable="true|false">
```

属性值

值	描述
true	规定元素可编辑。
false	规定元素不可编辑。

HTML 全局属性

HTML contextmenu 属性

HTML 全局属性

实例

规定 <div> 元素的上下文菜单。上下文菜单会在用户右键点击元素时出现：

```
<div contextmenu="mymenu">

<menu type="context" id="mymenu">
  <menuitem label="Refresh"></menuitem>
  <menuitem label="Twitter"></menuitem>
</menu>

</div>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前只有 Firefox 支持 contextmenu 属性。

定义和用法

contextmenu 属性规定元素的上下文菜单。当用户右键点击元素时，会出现上下文菜单。

contextmenu 属性的值是要打开的 <menu> 元素的 id。

HTML 4.01 与 HTML5 之间的差异

contextmenu 属性是 HTML5 中的新属性。

语法

```
<_element_ contextmenu="*menu_id*">
```

属性值

值	描述
menu_id	要打开的 <menu> 元素的 id。

HTML 全局属性

HTML data-* 属性

HTML 全局属性

实例

使用 data-* 属性来嵌入自定义数据：

```
<ul>
<li data-animal-type="bird">Owl</li>
<li data-animal-type="fish">Salmon</li>
<li data-animal-type="spider">Tarantula</li>
</ul>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 data-* 属性。

定义和用法

data-* 属性用于存储页面或应用程序的私有自定义数据。

data-* 属性赋予我们在所有 HTML 元素上嵌入自定义 data 属性的能力。

存储的（自定义）数据能够被页面的 JavaScript 中利用，以创建更好的用户体验（不进行 Ajax 调用或服务器端数据库查询）。

data-* 属性包括两部分：

- 属性名不应该包含任何大写字母，并且在前缀 "data-" 之后必须有至少一个字符
- 属性值可以是任意字符串

注释：用户代理会完全忽略前缀为 "data-" 的自定义属性。

HTML 4.01 与 HTML5 之间的差异

data-* 属性是 HTML5 中的新属性。

语法

```
<_element_ data-*="_somevalue_">
```

属性值

值	描述
somevalue	规定属性的值（以字符串）。

HTML 全局属性

HTML dir 属性

HTML 全局属性

实例

一段方向从右向左的段落：

```
<p dir="rtl">Write this text right-to-left!</p>
```

浏览器支持

所有浏览器均支持 dir 属性。

定义和用法

dir 属性规定元素内容的文本方向。

提示和注释

注释：dir 属性在以下标签中无效：<base>,
, <frame>, <frameset>, <hr>, <iframe>, <param> 以及 <script>。

语法

```
<element dir="ltr|rtl">
```

属性值

值	描述
ltr	默认。从左向右的文本方向。
rtl	从右向左的文本方向。

HTML 全局属性

HTML draggable 属性

HTML 全局属性

实例

一个可拖动的段落：

```
<p draggable="true">这是一个可拖动的段落。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 9+, Firefox, Opera, Chrome, and Safari 支持 draggable 属性。

注释：Internet Explorer 8 以及更早的版本，不支持 draggable 属性。

定义和用法

draggable 属性规定元素是否可拖动。

提示：链接和图像默认是可拖动的。

提示：draggable 属性常用在拖放操作中。请在我们的拖放教程中学习更多内容。

HTML 4.01 与 HTML5 之间的差异

draggable 属性是 HTML5 中的新属性。

语法

```
<_element_ draggable="true|false|auto">
```

属性值

值	描述
true	规定元素的可拖动的。
false	规定元素不可拖动。
auto	使用浏览器的默认行为。

HTML 全局属性

HTML dropzone 属性

HTML 全局属性

实例

拖动数据会产生被拖动数据的副本：

```
<div dropzone="copy"></div>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

目前所有主流浏览器都不支持 contenteditable 属性。

定义和用法

dropzone 属性规定在元素上拖动数据时，是否拷贝、移动或链接被拖动数据。

HTML 4.01 与 HTML5 之间的差异

dropzone 属性是 HTML5 中的新属性。

语法

```
<_element_ dropzone="copy|move|link">
```

属性值

值	描述
copy	拖动数据会产生被拖动数据的副本。
move	拖动数据会导致被拖动数据被移动到新位置。
link	拖动数据会产生指向原始数据的链接。

HTML 全局属性

HTML hidden 属性

HTML 全局属性

实例

被隐藏的段落：

```
<p hidden>这个段落应该被隐藏。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都支持 hidden 属性，除了 Internet Explorer。

定义和用法

hidden 属性是布尔属性。

如果设置该属性，它规定元素仍未或不再相关。

浏览器不应显示已规定 hidden 属性的元素。

hidden 属性也可用于防止用户查看元素，直到匹配某些条件（比如选择了某个复选框）。然后，JavaScript 可以删除 hidden 属性，以使此元素可见。

HTML 4.01 与 HTML5 之间的差异

hidden 属性是 HTML5 中的新属性。

HTML 与 XHTML 之间的差异

在 XHTML 中，属性简写是禁止的，hidden 属性必须定义为 <element hidden="hidden">。

语法

```
<_element_ hidden>
```

HTML 全局属性

HTML id 属性

HTML 全局属性

实例

通过 JavaScript 利用 id 属性来改变一段文本：

```
<html>
<head>
<script type="text/javascript">
function change_header()
{
document.getElementById("myHeader").innerHTML="Nice day!";
}
</script>
</head>

<body>
<h1 id="myHeader">Hello World!</h1>
<button onclick="change_header()">Change text</button>
</body>

</html>
```

（可以在本页底部找到更多实例）

定义和用法

id 属性规定 HTML 元素的唯一的 id。

id 在 HTML 文档中必须是唯一的。

id 属性可用作链接锚（link anchor），通过 JavaScript（HTML DOM）或通过 CSS 为带有指定 id 的元素改变或添加样式。

支持

W3C: "W3C" 列指示 W3C 的 HTML/XHTML 推荐标准中是否定义了该属性。

IE	Firefox	Opera	Safari	W3C
YES	YES	YES	YES	YES

语法

```
<element id="_value_">
```

属性值

值	描述
id	规定元素的唯一 id。

亲自试一试 - 实例

向 HTML 元素添加唯一的 id

如何向 HTML 元素添加 id 属性。

在 CSS 中使用 id 属性

如何在 CSS 中使用 id 属性

HTML 全局属性

HTML lang 属性

HTML 全局属性

实例

段落中的一些法文：

```
<p lang="fr">Ceci est un paragraphe.</p>
```

浏览器支持

所有浏览器均支持 lang 属性。

定义和用法

lang 属性规定元素内容的语言。

提示和注释

注释：lang 属性在以下标签中无效：<base>,
, <frame>, <frameset>, <hr>, <iframe>, <param> 以及 <script>。

语法

```
<element lang="*language_code*">
```

属性值

值	描述
language_code	规定元素内容的语言代码。语言代码参考

HTML 全局属性

HTML spellcheck 属性

HTML 全局属性

实例

进行拼写检查的可编辑段落：

```
<p contenteditable="true" spellcheck="true">这是一个段落。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

Internet Explorer 10, Firefox, Opera, Chrome 以及 Safari 支持 spellcheck 属性。

注释：Internet Explorer 9 以及更早的版本不支持 spellcheck 属性。

定义和用法

spellcheck 属性规定是否对元素进行拼写和语法检查。

可以对以下内容进行拼写检查：

- input 元素中的文本值（非密码）
- <textarea> 元素中的文本
- 可编辑元素中的文本

HTML 4.01 与 HTML5 之间的差异

spellcheck 属性是 HTML5 中的新属性。

语法

```
<_element_ spellcheck="true|false">
```

属性值

值	描述
true	对元素进行拼写和语法检查
false	不检查元素。

HTML 全局属性

HTML style 属性

HTML 全局属性

实例

在 HTML 文档中使用 style 属性：

```
<h1 style="color:blue; text-align:center">This is a header</h1>
<p style="color:red">This is a paragraph.</p>
```

（可以在本页底部找到更多实例）

定义和用法

style 属性规定元素的行内样式（inline style）

style 属性将覆盖任何全局的样式设定，例如在 <style> 标签或在外部样式表中规定的样式。

支持

W3C: "W3C" 列指示 W3C 的 HTML/XHTML 推荐标准中是否定义了该属性。

IE	Firefox	Opera	Safari	W3C
YES	YES	YES	YES	YES

语法

```
<element style="_value_">
```

属性值

值	描述
style_definition	一个或多个由分号分隔的 CSS 属性和值。

HTML 全局属性

HTML tabindex 属性

HTML 全局属性

实例

带有指定 tab 键顺序的链接：

```
<a href="http://www.w3school.com.cn/" tabindex="2">W3School</a>
<a href="http://www.google.com/" tabindex="1">Google</a>
<a href="http://www.microsoft.com/" tabindex="3">Microsoft</a>
```

浏览器支持

几乎所有浏览器均 tabindex 属性，除了 Safari。

定义和用法

tabindex 属性规定元素的 tab 键控制次序（当 tab 键用于导航时）。

提示和注释

注释：以下元素支持 tabindex 属性：<a>, <area>, <button>, <input>, <object>, <select> 以及 <textarea>。

语法

```
<element tabindex="_number_">
```

属性值

值	描述
number	规定元素的 tab 键控制次序（1 是第一个）。

HTML 全局属性

HTML title 属性

HTML 全局属性

实例

在 HTML 文档中使用 title 属性：

```
<abbr title="People's Republic of China">PRC</abbr> was founded in 1949.
<p title="Free Web tutorials">W3School.com.cn</p>
```

（可以在本页底部找到更多实例）

定义和用法

title 属性规定关于元素的额外信息。

这些信息通常会在鼠标移到元素上时显示一段工具提示文本（tooltip text）。

提示：title 属性常与 form 以及 a 元素一同使用，以提供关于输入格式和链接目标的信息。同时它也是 abbr 和 acronym 元素的必需属性。

支持

W3C: "W3C" 列指示 W3C 的 HTML/XHTML 推荐标准中是否定义了该属性。

IE	Firefox	Opera	Safari	W3C
YES	YES	YES	YES	YES

语法

```
<element title="_value_">
```

属性值

值	描述
text	规定元素的工具提示文本（tooltip text）。

HTML 全局属性

HTML translate 属性

HTML 全局属性

实例

规定不应翻译某些元素：

```
<p translate="no">请勿翻译本段。</p>
<p>本段可被译为任意语言。</p>
```

浏览器支持

IE	Firefox	Chrome	Safari	Opera

所有主流浏览器都无法正确地支持 translate 属性。

定义和用法

translate 规定是否应该翻译元素内容。

提示：请使用 class="notranslate" 替代。

HTML 4.01 与 HTML5 之间的差异

translate 属性是 HTML5 中的新属性。

语法

```
<_element_ translate="yes|no">
```

属性值

值	描述
yes	规定应该翻译元素内容。
no	规定不应翻译元素内容。

HTML 全局属性

HTML 事件属性

全局事件属性

HTML 4 增加了使事件在浏览器中触发动作的能力，比如当用户点击元素时启动 JavaScript。

如需学习更多有关事件编程的知识，请访问我们的 JavaScript 教程。

下面列出了添加到 HTML 元素以定义事件动作的全局事件属性。

= HTML5 中新的事件属性。

Window 事件属性

针对 window 对象触发的事件（应用到 <body> 标签）：

属性	值	描述
onafterprint	script	文档打印之后运行的脚本。
onbeforeprint	script	文档打印之前运行的脚本。
onbeforeunload	script	文档卸载之前运行的脚本。
onerror	script	在错误发生时运行的脚本。
onhaschange	script	当文档已改变时运行的脚本。
onload	script	页面结束加载之后触发。
onmessage	script	在消息被触发时运行的脚本。
onoffline	script	当文档离线时运行的脚本。
ononline	script	当文档上线时运行的脚本。
onpagehide	script	当窗口隐藏时运行的脚本。
onpageshow	script	当窗口成为可见时运行的脚本。
onpopstate	script	当窗口历史记录改变时运行的脚本。
onredo	script	当文档执行撤销（redo）时运行的脚本。
onresize	script	当浏览器窗口被调整大小时触发。
onstorage	script	在 Web Storage 区域更新后运行的脚本。
onundo	script	在文档执行 undo 时运行的脚本。
onunload	script	一旦页面已下载时触发（或者浏览器窗口已被关闭）。

Form 事件

由 HTML 表单内的动作触发的事件（应用到几乎所有 HTML 元素，但最常用在 form 元素中）：

属性	值	描述
onblur	script	元素失去焦点时运行的脚本。
onchange	script	在元素值被改变时运行的脚本。
oncontextmenu	script	当上下文菜单被触发时运行的脚本。
onfocus	script	当元素失去焦点时运行的脚本。
onformchange	script	在表单改变时运行的脚本。
onforminput	script	当表单获得用户输入时运行的脚本。
oninput	script	当元素获得用户输入时运行的脚本。
oninvalid	script	当元素无效时运行的脚本。
onreset	script	当表单中的重置按钮被点击时触发。HTML5 中不支持。
onselect	script	在元素中文本被选中后触发。
onsubmit	script	在提交表单时触发。

Keyboard 事件

属性	值	描述
onkeydown	script	在用户按下按键时触发。
onkeypress	script	在用户敲击按钮时触发。
onkeyup	script	当用户释放按键时触发。

Mouse 事件

由鼠标或类似用户动作触发的事件：

属性	值	描述
onclick	script	元素上发生鼠标点击时触发。
ondblclick	script	元素上发生鼠标双击时触发。
ondrag	script	元素被拖动时运行的脚本。
ondragend	script	在拖动操作末端运行的脚本。
ondragenter	script	当元素元素已被拖动到有效拖放区域时运行的脚本。
ondragleave	script	当元素离开有效拖放目标时运行的脚本。
ondragover	script	当元素在有效拖放目标上正在被拖动时运行的脚本。
ondragstart	script	在拖动操作开端运行的脚本。
ondrop	script	当被拖元素正在被拖放时运行的脚本。
onmousedown	script	当元素上按下鼠标按钮时触发。
onmousemove	script	当鼠标指针移动到元素上时触发。
onmouseout	script	当鼠标指针移出元素时触发。
onmouseover	script	当鼠标指针移动到元素上时触发。
onmouseup	script	当在元素上释放鼠标按钮时触发。
onmousewheel	script	当鼠标滚轮正在被滚动时运行的脚本。
onscroll	script	当元素滚动条被滚动时运行的脚本。

Media 事件

由媒介（比如视频、图像和音频）触发的事件（适用于所有 HTML 元素，但常见于媒介元素中，比如 <audio>、<embed>、<img>、<object> 以及 <video>）：

属性	值	描述
onabort	script	在退出时运行的脚本。
oncanplay	script	当文件就绪可以开始播放时运行的脚本（缓冲已足够开始时）。
oncanplaythrough	script	当媒介能够无需因缓冲而停止即可播放至结尾时运行的脚本。
ondurationchange	script	当媒介长度改变时运行的脚本。
onemptied	script	当发生故障并且文件突然不可用时运行的脚本（比如连接意外断开时）。
onended	script	当媒介已到达结尾时运行的脚本（可发送类似“感谢观看”之类的消息）。

onerror	script	当在文件加载期间发生错误时运行的脚本。
onloadeddata	script	当媒介数据已加载时运行的脚本。
onloadedmetadata	script	当元数据（比如分辨率和时长）被加载时运行的脚本。
onloadstart	script	在文件开始加载且未实际加载任何数据前运行的脚本。
onpause	script	当媒介被用户或程序暂停时运行的脚本。
onplay	script	当媒介已就绪可以开始播放时运行的脚本。
onplaying	script	当媒介已开始播放时运行的脚本。
onprogress	script	当浏览器正在获取媒介数据时运行的脚本。
onratechange	script	每当回放速率改变时运行的脚本（比如当用户切换到慢动作或快进模式）。
onreadystatechange	script	每当就绪状态改变时运行的脚本（就绪状态监测媒介数据的状态）。
onseeked	script	当 seeking 属性设置为 false（指示定位已结束）时运行的脚本。
onseeking	script	当 seeking 属性设置为 true（指示定位是活动的）时运行的脚本。
onstalled	script	在浏览器不论何种原因未能取回媒介数据时运行的脚本。
onsuspend	script	在媒介数据完全加载之前不论何种原因终止取回媒介数据时运行的脚本。
ontimeupdate	script	当播放位置改变时（比如当用户快进到媒介中一个不同的位置时）运行的脚本。
onvolumechange	script	每当音量改变时（包括将音量设置为静音）时运行的脚本。
onwaiting	script	当媒介已停止播放但打算继续播放时（比如当媒介暂停已缓冲更多数据）运行脚本

HTML 5 视频/音频参考手册

HTML5 视频和音频的 DOM 参考手册

HTML5 DOM 为 <audio> 和 <video> 元素提供了方法、属性和事件。

这些方法、属性和事件允许您使用 JavaScript 来操作 <audio> 和 <video> 元素。

HTML5 Audio/Video 方法

方法	描述
addTextTrack()	向音频/视频添加新的文本轨道
canPlayType()	检测浏览器是否能播放指定的音频/视频类型
load()	重新加载音频/视频元素
play()	开始播放音频/视频
pause()	暂停当前播放的音频/视频

HTML5 Audio/Video 属性

属性	描述
audioTracks	返回表示可用音轨的 AudioTrackList 对象
autoplay	设置或返回是否在加载完成后随即播放音频/视频
buffered	返回表示音频/视频已缓冲部分的 TimeRanges 对象
controller	返回表示音频/视频当前媒体控制器的 MediaController 对象
controls	设置或返回音频/视频是否显示控件（比如播放/暂停等）
crossOrigin	设置或返回音频/视频的 CORS 设置
currentSrc	返回当前音频/视频的 URL
currentTime	设置或返回音频/视频中的当前播放位置（以秒计）
defaultMuted	设置或返回音频/视频默认是否静音
defaultPlaybackRate	设置或返回音频/视频的默认播放速度
duration	返回当前音频/视频的长度（以秒计）
ended	返回音频/视频的播放是否已结束
error	返回表示音频/视频错误状态的 MediaError 对象
loop	设置或返回音频/视频是否应在结束时重新播放
mediaGroup	设置或返回音频/视频所属的组合（用于连接多个音频/视频元素）
muted	设置或返回音频/视频是否静音
networkState	返回音频/视频的当前网络状态
paused	设置或返回音频/视频是否暂停
playbackRate	设置或返回音频/视频播放的速度
played	返回表示音频/视频已播放部分的 TimeRanges 对象
preload	设置或返回音频/视频是否应该在页面加载后进行加载
readyState	返回音频/视频当前的就绪状态
seekable	返回表示音频/视频可寻址部分的 TimeRanges 对象
seeking	返回用户是否正在音频/视频中进行查找
src	设置或返回音频/视频元素的当前来源
startDate	返回表示当前时间偏移的 Date 对象
textTracks	返回表示可用文本轨道的 TextTrackList 对象
videoTracks	返回表示可用视频轨道的 VideoTrackList 对象
volume	设置或返回音频/视频的音量

HTML5 Audio/Video 事件

事件	描述
abort	当音频/视频的加载已放弃时
canplay	当浏览器可以播放音频/视频时
canplaythrough	当浏览器可在不因缓冲而停顿的情况下进行播放时
durationchange	当音频/视频的时长已更改时
emptied	当目前的播放列表为空时
ended	当目前的播放列表已结束时
error	当在音频/视频加载期间发生错误时
loadeddata	当浏览器已加载音频/视频的当前帧时
loadedmetadata	当浏览器已加载音频/视频的元数据时
loadstart	当浏览器开始查找音频/视频时
pause	当音频/视频已暂停时
play	当音频/视频已开始或不再暂停时
playing	当音频/视频在已因缓冲而暂停或停止后已就绪时
progress	当浏览器正在下载音频/视频时
ratechange	当音频/视频的播放速度已更改时
seeked	当用户已移动/跳跃到音频/视频中的新位置时
seeking	当用户开始移动/跳跃到音频/视频中的新位置时
stalled	当浏览器尝试获取媒体数据，但数据不可用时
suspend	当浏览器刻意不获取媒体数据时
timeupdate	当目前的播放位置已更改时
volumechange	当音量已更改时
waiting	当视频由于需要缓冲下一帧而停止

HTML 5 Canvas 参考手册

描述

HTML5 <canvas> 标签用于绘制图像（通过脚本，通常是 JavaScript）。

不过，<canvas> 元素本身并没有绘制能力（它仅仅是图形的容器） - 您必须使用脚本来完成实际的绘图任务。

getContext() 方法可返回一个对象，该对象提供了用于在画布上绘图的方法和属性。

本手册提供完整的 getContext("2d") 对象属性和方法，可用于在画布上绘制文本、线条、矩形、圆形等等。

浏览器支持

Internet Explorer 9、Firefox、Opera、Chrome 以及 Safari 支持 <canvas> 及其属性和方法。

注释：Internet Explorer 8 以及更早的版本不支持 <canvas> 元素。

颜色、样式和阴影

属性	描述
fillStyle	设置或返回用于填充绘画的颜色、渐变或模式
strokeStyle	设置或返回用于笔触的颜色、渐变或模式
shadowColor	设置或返回用于阴影的颜色
shadowBlur	设置或返回用于阴影的模糊级别
shadowOffsetX	设置或返回阴影距形状的水平距离
shadowOffsetY	设置或返回阴影距形状的垂直距离

方法	描述
createLinearGradient()	创建线性渐变（用在画布内容上）
createPattern()	在指定的方向上重复指定的元素
createRadialGradient()	创建放射状/环形的渐变（用在画布内容上）
addColorStop()	规定渐变对象中的颜色和停止位置

线条样式

属性	描述
lineCap	设置或返回线条的结束端点样式
lineJoin	设置或返回两条线相交时，所创建的拐角类型
lineWidth	设置或返回当前的线条宽度
miterLimit	设置或返回最大斜接长度

矩形

方法	描述
rect()	创建矩形
fillRect()	绘制“被填充”的矩形
strokeRect()	绘制矩形（无填充）
clearRect()	在给定的矩形内清除指定的像素

路径

方法	描述
fill()	填充当前绘图（路径）
stroke()	绘制已定义的路径
beginPath()	起始一条路径，或重置当前路径
moveTo()	把路径移动到画布中的指定点，不创建线条
closePath()	创建从当前点回到起始点的路径
lineTo()	添加一个新点，然后在画布中创建从该点到最后指定点的线条
clip()	从原始画布剪切任意形状和尺寸的区域
quadraticCurveTo()	创建二次贝塞尔曲线
bezierCurveTo()	创建三次方贝塞尔曲线
arc()	创建弧/曲线（用于创建圆形或部分圆）
arcTo()	创建两切线之间的弧/曲线
isPointInPath()	如果指定的点位于当前路径中，则返回 true，否则返回 false

转换

方法	描述
scale()	缩放当前绘图至更大或更小
rotate()	旋转当前绘图
translate()	重新映射画布上的 (0,0) 位置
transform()	替换绘图的当前转换矩阵
setTransform()	将当前转换重置为单位矩阵。然后运行 transform()

文本

属性	描述
font	设置或返回文本内容的当前字体属性
textAlign	设置或返回文本内容的当前对齐方式
textBaseline	设置或返回在绘制文本时使用的当前文本基线

方法	描述
fillText()	在画布上绘制“被填充的”文本
strokeText()	在画布上绘制文本（无填充）
measureText()	返回包含指定文本宽度的对象

图像绘制

方法	描述
drawImage()	向画布上绘制图像、画布或视频

像素操作

属性	描述
width	返回 ImageData 对象的宽度
height	返回 ImageData 对象的高度
data	返回一个对象，其包含指定的 ImageData 对象的图像数据

方法	描述
createImageData()	创建新的、空白的 ImageData 对象
getImageData()	返回 ImageData 对象，该对象为画布上指定的矩形复制像素数据
putImageData()	把图像数据（从指定的 ImageData 对象）放回画布上

合成

属性	描述
globalAlpha	设置或返回绘图的当前 alpha 或透明值
globalCompositeOperation	设置或返回新图像如何绘制到已有的图像上

其他

方法	描述
save()	保存当前环境的状态
restore()	返回之前保存过的路径状态和属性
createEvent()	
getContext()	
toDataURL()	

HTML 元素和有效的 DTD

HTML5/HTML 4.01/XHTML 元素和有效的 DTD

下面的表格列出了所有的 HTML5/HTML 4.01/XHTML 元素，以及它们会出现在什么文档类型 (DTD) 中：

标签	HTML5	HTML 4.01 / XHTML 1.0	XHTML 1.1		
Transitional	Strict	Frameset			
---	---	---			
<a>	Yes	Yes	Yes	Yes	Yes
<abbr>	Yes	Yes	Yes	Yes	Yes
<acronym>	No	Yes	Yes	Yes	Yes
<address>	Yes	Yes	Yes	Yes	Yes
<applet>	No	Yes	No	Yes	No
<area>	Yes	Yes	Yes	Yes	No
<article>	Yes	No	No	No	No
<aside>	Yes	No	No	No	No
<audio>	Yes	No	No	No	No
<b>	Yes	Yes	Yes	Yes	Yes
<base>	Yes	Yes	Yes	Yes	Yes
<basefont>	No	Yes	No	Yes	No
<bdi>	Yes	No	No	No	No
<bdo>	Yes	Yes	Yes	Yes	No
<big>	No	Yes	Yes	Yes	Yes
<blockquote>	Yes	Yes	Yes	Yes	Yes
<body>	Yes	Yes	Yes	Yes	Yes
 	Yes	Yes	Yes	Yes	Yes
<button>	Yes	Yes	Yes	Yes	Yes
<canvas>	Yes	No	No	No	No
<caption>	Yes	Yes	Yes	Yes	Yes
<center>	No	Yes	No	Yes	No

<cite>	Yes	Yes	Yes	Yes	Yes
<code>	Yes	Yes	Yes	Yes	Yes
<col>	Yes	Yes	Yes	Yes	No
<colgroup>	Yes	Yes	Yes	Yes	No
<command>	Yes	No	No	No	No
<datalist>	Yes	No	No	No	No
<dd>	Yes	Yes	Yes	Yes	Yes
<del>	Yes	Yes	Yes	Yes	No
<details>	Yes	No	No	No	No
<dfn>	Yes	Yes	Yes	Yes	Yes
<dir>	No	Yes	No	Yes	No
<div>	Yes	Yes	Yes	Yes	Yes
<dl>	Yes	Yes	Yes	Yes	Yes
<dt>	Yes	Yes	Yes	Yes	Yes
<em>	Yes	Yes	Yes	Yes	Yes
<embed>	Yes	No	No	No	No
<fieldset>	Yes	Yes	Yes	Yes	Yes
<figcaption>	Yes	No	No	No	No
<figure>	Yes	No	No	No	No
<font>	No	Yes	No	Yes	No
<footer>	Yes	No	No	No	No
<form>	Yes	Yes	Yes	Yes	Yes
<frame>	No	No	No	Yes	No
<frameset>	No	No	No	Yes	No
<head>	Yes	Yes	Yes	Yes	Yes
<header>	Yes	No	No	No	No
<h1> to <h6>	Yes	Yes	Yes	Yes	Yes
<hr>	Yes	Yes	Yes	Yes	Yes
<html>	Yes	Yes	Yes	Yes	Yes
<i>	Yes	Yes	Yes	Yes	Yes
<iframe>	Yes	Yes	No	Yes	No

<img>	Yes	Yes	Yes	Yes	Yes
<input>	Yes	Yes	Yes	Yes	Yes
<ins>	Yes	Yes	Yes	Yes	No
<keygen>	Yes	No	No	No	No
<kbd>	Yes	Yes	Yes	Yes	Yes
<label>	Yes	Yes	Yes	Yes	Yes
<legend>	Yes	Yes	Yes	Yes	Yes
	Yes	Yes	Yes	Yes	Yes
<link>	Yes	Yes	Yes	Yes	Yes
<map>	Yes	Yes	Yes	Yes	No
<mark>	Yes	No	No	No	No
<menu>	Yes	Yes	No	Yes	No
<meta>	Yes	Yes	Yes	Yes	Yes
<meter>	Yes	No	No	No	No
<nav>	Yes	No	No	No	No
<noframes>	No	Yes	No	Yes	No
<noscript>	Yes	Yes	Yes	Yes	Yes
<object>	Yes	Yes	Yes	Yes	Yes
	Yes	Yes	Yes	Yes	Yes
<optgroup>	Yes	Yes	Yes	Yes	Yes
<option>	Yes	Yes	Yes	Yes	Yes
<output>	Yes	No	No	No	No
<p>	Yes	Yes	Yes	Yes	Yes
<param>	Yes	Yes	Yes	Yes	Yes
<pre>	Yes	Yes	Yes	Yes	Yes
<progress>	Yes	No	No	No	No
<q>	Yes	Yes	Yes	Yes	Yes
<rp>	Yes	No	No	No	No
<rt>	Yes	No	No	No	No
<ruby>	Yes	No	No	No	No
<s>	Yes	Yes	No	Yes	No
<samp>	Yes	Yes	Yes	Yes	Yes

<script>	Yes	Yes	Yes	Yes	Yes
<section>	Yes	No	No	No	No
<select>	Yes	Yes	Yes	Yes	Yes
<small>	Yes	Yes	Yes	Yes	Yes
<source>	Yes	No	No	No	No
<span>	Yes	Yes	Yes	Yes	Yes
<strike>	No	Yes	No	Yes	No
<strong>	Yes	Yes	Yes	Yes	Yes
<style>	Yes	Yes	Yes	Yes	Yes
<sub>	Yes	Yes	Yes	Yes	Yes
<summary>	Yes	No	No	No	No
<sup>	Yes	Yes	Yes	Yes	Yes
<table>	Yes	Yes	Yes	Yes	Yes
<tbody>	Yes	Yes	Yes	Yes	No
<td>	Yes	Yes	Yes	Yes	Yes
<textarea>	Yes	Yes	Yes	Yes	Yes
<tfoot>	Yes	Yes	Yes	Yes	No
<th>	Yes	Yes	Yes	Yes	Yes
<thead>	Yes	Yes	Yes	Yes	No
<time>	Yes	No	No	No	No
<title>	Yes	Yes	Yes	Yes	Yes
<tr>	Yes	Yes	Yes	Yes	Yes
<track>	Yes	No	No	No	No
<tt>	No	Yes	Yes	Yes	Yes
<u>	No	Yes	No	Yes	No
	Yes	Yes	Yes	Yes	Yes
<var>	Yes	Yes	Yes	Yes	Yes
<video>	Yes	No	No	No	No
<wbr>	Yes	No	No	No	No

HTML 颜色名

本页提供了被大多数浏览器支持的颜色名。

提示：W3C 的 HTML 4.0 标准仅支持 16 种颜色名，它们是：aqua、black、blue、fuchsia、gray、green、lime、maroon、navy、olive、purple、red、silver、teal、white、yellow。

如果使用其它颜色的话，就应该使用十六进制的颜色值。

颜色名列表

单击一个颜色名或者 16 进制值，就可以查看与不同文字颜色搭配的背景颜色。

颜色名	十六进制颜色值	颜色
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
Coral	#FF7F50	
CornflowerBlue	#6495ED	
Cornsilk	#FFF8DC	
Crimson	#DC143C	

Cyan	#00FFFF
DarkBlue	#00008B
DarkCyan	#008B8B
DarkGoldenRod	#B8860B
DarkGray	#A9A9A9
DarkGreen	#006400
DarkKhaki	#BDB76B
DarkMagenta	#8B008B
DarkOliveGreen	#556B2F
Darkorange	#FF8C00
DarkOrchid	#9932CC
DarkRed	#8B0000
DarkSalmon	#E9967A
DarkSeaGreen	#8FBC8F
DarkSlateBlue	#483D8B
DarkSlateGray	#2F4F4F
DarkTurquoise	#00CED1
DarkViolet	#9400D3
DeepPink	#FF1493
DeepSkyBlue	#00BFFF
DimGray	#696969
DodgerBlue	#1E90FF
Feldspar	#D19275
FireBrick	#B22222
FloralWhite	#FFFAF0
ForestGreen	#228B22
Fuchsia	#FF00FF
Gainsboro	#DCDCDC
GhostWhite	#F8F8FF
Gold	#FFD700
GoldenRod	#DAA520

Gray	#808080
Green	#008000
GreenYellow	#ADFF2F
HoneyDew	#F0FFF0
HotPink	#FF69B4
IndianRed	#CD5C5C
Indigo	#4B0082
Ivory	#FFFFF0
Khaki	#F0E68C
Lavender	#E6E6FA
LavenderBlush	#FFF0F5
LawnGreen	#7CFC00
LemonChiffon	#FFFACD
LightBlue	#ADD8E6
LightCoral	#F08080
LightCyan	#E0FFFF
LightGoldenRodYellow	#FAFAD2
LightGrey	#D3D3D3
LightGreen	#90EE90
LightPink	#FFB6C1
LightSalmon	#FFA07A
LightSeaGreen	#20B2AA
LightSkyBlue	#87CEFA
LightSlateBlue	#8470FF
LightSlateGray	#778899
LightSteelBlue	#B0C4DE
LightYellow	#FFFFE0
Lime	#00FF00
LimeGreen	#32CD32
Linen	#FAF0E6
Magenta	#FF00FF
Maroon	#800000

MediumAquaMarine	#66CDAA
MediumBlue	#0000CD
MediumOrchid	#BA55D3
MediumPurple	#9370D8
MediumSeaGreen	#3CB371
MediumSlateBlue	#7B68EE
MediumSpringGreen	#00FA9A
MediumTurquoise	#48D1CC
MediumVioletRed	#C71585
MidnightBlue	#191970
MintCream	#F5FFFA
MistyRose	#FFE4E1
Moccasin	#FFE4B5
NavajoWhite	#FFDEAD
Navy	#000080
OldLace	#FDF5E6
Olive	#808000
OliveDrab	#6B8E23
Orange	#FFA500
OrangeRed	#FF4500
Orchid	#DA70D6
PaleGoldenRod	#EEE8AA
PaleGreen	#98FB98
PaleTurquoise	#AFEEEE
PaleVioletRed	#D87093
PapayaWhip	#FFEFD5
PeachPuff	#FFDAB9
Peru	#CD853F
Pink	#FFC0CB
Plum	#DDA0DD
PowderBlue	#B0E0E6

Purple	#800080
Red	#FF0000
RosyBrown	#BC8F8F
RoyalBlue	#4169E1
SaddleBrown	#8B4513
Salmon	#FA8072
SandyBrown	#F4A460
SeaGreen	#2E8B57
SeaShell	#FFF5EE
Sienna	#A0522D
Silver	#C0C0C0
SkyBlue	#87CEEB
SlateBlue	#6A5ACD
SlateGray	#708090
Snow	#FFFAFA
SpringGreen	#00FF7F
SteelBlue	#4682B4
Tan	#D2B48C
Teal	#008080
Thistle	#D8BFD8
Tomato	#FF6347
Turquoise	#40E0D0
Violet	#EE82EE
VioletRed	#D02090
Wheat	#F5DEB3
White	#FFFFFF
WhiteSmoke	#F5F5F5
Yellow	#FFFF00
YellowGreen	#9ACD32

HTML 字符集

HTML 字符集

如需正确地显示 HTML 页面，浏览器必须知道使用何种字符集。

万维网早期使用的字符集是 ASCII。ASCII 支持 0-9 的数字，大写和小写英文字母表，以及一些特殊字符。

完整的 ASCII 参考手册

由于很多国家使用的字符并不属于 ASCII，现代浏览器的默认字符集是 ISO-8859-1。

完整的 ISO-8859-1 参考手册

如果网页使用不同于 ISO-8859-1 的字符集，就应该在 <meta> 标签进行指定。

ISO 字符集

ISO 字符集是国际标准组织 (ISO) 针对不同的字母表/语言定义的标准字符集。

下面列出了世界各地使用的不同字符集：

字符集	描述	使用范围
ISO-8859-1	Latin alphabet part 1	北美、西欧、拉丁美洲、加勒比海、加拿大、非洲
ISO-8859-2	Latin alphabet part 2	东欧
ISO-8859-3	Latin alphabet part 3	SE Europe、世界语、其他杂项
ISO-8859-4	Latin alphabet part 4	斯堪的纳维亚/波罗的海（以及其他没有包括在 ISO-8859-1 中的部分）
ISO-8859-5	Latin/Cyrillic part 5	使用古代斯拉夫语字母表的语言，比如保加利亚语、白俄罗斯文、俄罗斯语、马其顿语
ISO-8859-6	Latin/Arabic part 6	使用阿拉伯字母的语言
ISO-8859-7	Latin/Greek part 7	现代希腊语，以及有希腊语衍生的数学符号
ISO-8859-8	Latin/Hebrew part 8	使用希伯来语的语言
ISO-8859-9	Latin 5 part 9	土耳其语
ISO-8859-10	Latin 6	拉普兰语、日耳曼语、爱斯基摩北欧语
ISO-8859-15	Latin 9 (aka Latin 0)	与 ISO 8859-1 类似，欧元符号和其他一些字符取代了一些较少使用的符号
ISO-2022-JP	Latin/Japanese part 1	日本语
ISO-2022-JP-2	Latin/Japanese part 2	日本语
ISO-2022-KR	Latin/Korean part 1	韩语

Unicode 标准

由于上面列出的字符集都有容量限制，而且不兼容多语言环境，Unicode 联盟开发了 Unicode 标准。

Unicode 标准涵盖了世界上的所有字符、标点和符号。

不论是何种平台、程序或语言，Unicode 都能够进行文本数据的处理、存储和交换。

Unicode 联盟

Unicode 联盟开发了 Unicode 标准。他们的目标是用标准的 Unicode 转换格式 (UTF) 来取代现有的字符集。

Unicode 标准已经获得了成功，在 XML、Java、ECMAScript (JavaScript)、LDAP、CORBA 3.0、WML 中，Unicode 已经得到了实现。在许多操作系统以及所有的现代浏览器中，Unicode 同样得到了支持。

Unicode 联盟与领导性的标准发展组织进行合作，比如 ISO、W3C 以及 ECMA。

Unicode 可以被不同的字符集兼容。最常用的编码方式是 UTF-8 和 UTF-16：

UTF-8

UTF8 中的字符可以是 1-4 个字节长。UTF-8 可以表示 Unicode 标准中的任意字符。UTF-8 向后兼容 ASCII。UTF-8 是网页和电子邮件的首选编码。

UTF-16

16 比特的 Unicode 转换格式是一种 Unicode 可变字符编码，能够对全部 Unicode 指令表进行编码。UTF-16 主要被用于操作系统和环境中，比如微软的 Windows 2000/XP/2003/Vista/CE 以及 Java 和 .NET 字节代码环境。

提示：最前面的 256 个 Unicode 字符集字符对应于 256 个 ISO-8859-1 字符。

提示：所有 HTML 4 处理器均已支持 UTF-8，而所有 XHTML 和 XML 处理器支持 UTF-8 和 UTF-16。

HTML ASCII 参考手册

HTML 和 XHTML 用标准的 7 比特 ASCII 代码在网络上传输数据。

7 比特 ASCII 代码可提供 128 个不同的字符值。

7 比特 可显示的 ASCII 代码

结果	描述	实体编号
space		
!	exclamation mark	!
"	quotation mark	"
#	number sign	#
$	dollar sign	$
%	percent sign	%
&	ampersand	&
'	apostrophe	'
(	left parenthesis	(
)	right parenthesis	)
*	asterisk	*
+	plus sign	+
,	comma	,
-	hyphen	-
.	period	.
/	slash	/
0	digit 0	0
1	digit 1	1
2	digit 2	2
3	digit 3	3
4	digit 4	4
5	digit 5	5
6	digit 6	6

7	digit 7	7
8	digit 8	8
9	digit 9	9
:	colon	:
;	semicolon	;
<	less-than	<
=	equals-to	=
>	greater-than	>
?	question mark	?
@	at sign	@
A	uppercase A	A
B	uppercase B	B
C	uppercase C	C
D	uppercase D	D
E	uppercase E	E
F	uppercase F	F
G	uppercase G	G
H	uppercase H	H
I	uppercase I	I
J	uppercase J	J
K	uppercase K	K
L	uppercase L	L
M	uppercase M	M
N	uppercase N	N
O	uppercase O	O
P	uppercase P	P
Q	uppercase Q	Q
R	uppercase R	R
S	uppercase S	S
T	uppercase T	T
U	uppercase U	U
V	uppercase V	V

W	uppercase W	W
X	uppercase X	X
Y	uppercase Y	Y
Z	uppercase Z	Z
[	left square bracket	[
\	backslash	\
]	right square bracket	]
^	caret	^
_	underscore	_
`	grave accent	`
a	lowercase a	a
b	lowercase b	b
c	lowercase c	c
d	lowercase d	d
e	lowercase e	e
f	lowercase f	f
g	lowercase g	g
h	lowercase h	h
i	lowercase i	i
j	lowercase j	j
k	lowercase k	k
l	lowercase l	l
m	lowercase m	m
n	lowercase n	n
o	lowercase o	o
p	lowercase p	p
q	lowercase q	q
r	lowercase r	r
s	lowercase s	s
t	lowercase t	t
u	lowercase u	u

v	lowercase v	v
w	lowercase w	w
x	lowercase x	x
y	lowercase y	y
z	lowercase z	z
{	left curly brace	{
		vertical bar		
}	right curly brace	}
~	tilde	~

7 比特 设备控制 ASCII代码

ASCII设备控制代码最初被设计为用来控制诸如打印机和磁带驱动器之类的硬件设备。在HTML文档中这些代码不会起任何作用。

结果	描述	实体编号
NUL	null character	&#00;
SOH	start of header	&#01;
STX	start of text	&#02;
ETX	end of text	&#03;
EOT	end of transmission	&#04;
ENQ	enquiry	&#05;
ACK	acknowledge	&#06;
BEL	bell (ring)	&#07;
BS	backspace	&#08;
HT	horizontal tab	
LF	line feed
VT	vertical tab	&#11;
FF	form feed	
CR	carriage return
SO	shift out	&#14;
SI	shift in	&#15;
DLE	data link escape	&#16;

DC1	device control 1	&#17;
DC2	device control 2	&#18;
DC3	device control 3	&#19;
DC4	device control 4	&#20;
NAK	negative acknowledge	&#21;
SYN	synchronize	&#22;
ETB	end transmission block	&#23;
CAN	cancel	&#24;
EM	end of medium	&#25;
SUB	substitute	&#26;
ESC	escape	&#27;
FS	file separator	&#28;
GS	group separator	&#29;
RS	record separator	&#30;
US	unit separator	&#31;
DEL	delete (rubout)	&#127;

HTML ISO-8859-1 参考手册

HTML 4.01 支持 ISO 8859-1 (Latin-1) 字符集。

ISO-8859-1 的较低部分（从 1 到 127 之间的代码）是最初的 7 比特 ASCII。

ISO-8859-1 的较高部分（从 160 到 255 之间的代码）全都有实体名称。

这些符号中的大多数都可以在不进行实体引用的情况下使用，但是实体名称或实体编号为那些不容易通过键盘键入的符号提供了表达的方法。

注释：实体名称对大小写敏感。

带有实体名称的 ASCII 实体

结果	描述	实体名称	实体编号
"	quotation mark	"	"
'	apostrophe	'	'
&	ampersand	&	&
<	less-than	<	<
>	greater-than	>	>

ISO 8859-1 符号实体

结果	描述	实体名称	实体编号
	non-breaking space		
?	inverted exclamation mark	¡	¡
¢	cent	¢	¢
£	pound	£	£
¤	currency	¤	¤
¥	yen	¥	¥
|	broken vertical bar	¦	¦
§	section	§	§
¨	spacing diaeresis	¨	¨
?	copyright	©	©

a	feminine ordinal indicator	ª	ª
?	angle quotation mark (left)	«	«
?	negation	¬	¬
-	soft hyphen		
?	registered trademark	®	®
-	spacing macron	¯	¯
°	degree	°	°
±	plus-or-minus	±	±
2	superscript 2	²	²
3	superscript 3	³	³
′	spacing acute	´	´
µ	micro	µ	µ
?	paragraph	¶	¶
·	middle dot	·	·
?	spacing cedilla	¸	¸
1	superscript 1	¹	¹
o	masculine ordinal indicator	º	º
?	angle quotation mark (right)	»	»
?	fraction 1/4	¼	¼
?	fraction 1/2	½	½
?	fraction 3/4	¾	¾
?	inverted question mark	¿	¿
×	multiplication	×	×
÷	division	÷	÷

ISO 8859-1 字符实体

结果	描述	实体名称	实体编号
à	capital a, grave accent	À	À
á	capital a, acute accent	Á	Á
?	capital a, circumflex accent	Â	Â
?	capital a, tilde	Ã	Ã

?	capital a, umlaut mark	Ä	Ä
?	capital a, ring	Å	Å
?	capital ae	Æ	Æ
?	capital c, cedilla	Ç	Ç
è	capital e, grave accent	È	È
é	capital e, acute accent	É	É
ê	capital e, circumflex accent	Ê	Ê
?	capital e, umlaut mark	Ë	Ë
ì	capital i, grave accent	Ì	Ì
í	capital i, acute accent	Í	Í
?	capital i, circumflex accent	Î	Î
?	capital i, umlaut mark	Ï	Ï
D	capital eth, Icelandic	Ð	Ð
?	capital n, tilde	Ñ	Ñ
ò	capital o, grave accent	Ò	Ò
ó	capital o, acute accent	Ó	Ó
?	capital o, circumflex accent	Ô	Ô
?	capital o, tilde	Õ	Õ
?	capital o, umlaut mark	Ö	Ö
?	capital o, slash	Ø	Ø
ù	capital u, grave accent	Ù	Ù
ú	capital u, acute accent	Ú	Ú
?	capital u, circumflex accent	Û	Û
ü	capital u, umlaut mark	Ü	Ü
Y	capital y, acute accent	Ý	Ý
T	capital THORN, Icelandic	Þ	Þ
?	small sharp s, German	ß	ß
à	small a, grave accent	à	à
á	small a, acute accent	á	á
a	small a, circumflex accent	â	â
?	small a, tilde	ã	ã
?	small a, umlaut mark	ä	ä

?	small a, ring	å	å
?	small ae	æ	æ
?	small c, cedilla	ç	ç
è	small e, grave accent	è	è
é	small e, acute accent	é	é
ê	small e, circumflex accent	ê	ê
?	small e, umlaut mark	ë	ë
ì	small i, grave accent	ì	ì
í	small i, acute accent	í	í
?	small i, circumflex accent	î	î
?	small i, umlaut mark	ï	ï
e	small eth, Icelandic	ð	ð
?	small n, tilde	ñ	ñ
ò	small o, grave accent	ò	ò
ó	small o, acute accent	ó	ó
?	small o, circumflex accent	ô	ô
?	small o, tilde	õ	õ
?	small o, umlaut mark	ö	ö
?	small o, slash	ø	ø
ù	small u, grave accent	ù	ù
ú	small u, acute accent	ú	ú
?	small u, circumflex accent	û	û
ü	small u, umlaut mark	ü	ü
y	small y, acute accent	ý	ý
t	small thorn, Icelandic	þ	þ
?	small y, umlaut mark	ÿ	ÿ

HTML 4.01 符号实体

本字符实体参考手册包括了数学符号、希腊字符、各种箭头记号、科技符号以及形状。

注释：实体名称对大小写敏感。

HTML 支持的数学符号

结果	描述	实体名称	实体编号
?	for all	∀	∀
?	part	∂	∂
?	exists	&exists;	∃
?	empty	∅	∅
?	nabla	∇	∇
∈	isin	∈	∈
?	notin	∉	∉
?	ni	∋	∋
∏	prod	∏	∏
∑	sum	∑	∑
?	minus	−	−
?	lowast	∗	∗
√	square root	√	√
∝	proportional to	∝	∝
∞	infinity	∞	∞
∠	angle	∠	∠
∧	and	∧	∧
∨	or	∨	∨
∩	cap	∩	∩
∪	cup	∪	∪
∫	integral	∫	∫
∴	therefore	∴	∴
∼	simular to	∼	∼

?	approximately equal	≅	≅
≈	almost equal	≈	≈
≠	not equal	≠	≠
≡	equivalent	≡	≡
≤	less or equal	≤	≤
≥	greater or equal	≥	≥
?	subset of	⊂	⊂
?	superset of	⊃	⊃
?	not subset of	⊄	⊄
?	subset or equal	⊆	⊆
?	superset or equal	⊇	⊇
⊕	circled plus	⊕	⊕
?	cirled times	⊗	⊗
⊥	perpendicular	⊥	⊥
?	dot operator	⋅	⋅

HTML 支持的希腊字母

结果	描述	实体名称	实体编号
Α	Alpha	Α	Α
Β	Beta	Β	Β
Γ	Gamma	Γ	Γ
Δ	Delta	Δ	Δ
Ε	Epsilon	Ε	Ε
Ζ	Zeta	Ζ	Ζ
Η	Eta	Η	Η
Θ	Theta	Θ	Θ
Ι	Iota	Ι	Ι
Κ	Kappa	Κ	Κ
Λ	Lambda	Λ	Λ
Μ	Mu	Μ	Μ
Ν	Nu	Ν	Ν

Ξ	Xi	Ξ	Ξ
Ο	Omicron	Ο	Ο
Π	Pi	Π	Π
Ρ	Rho	Ρ	Ρ
Sigmaf	undefined		
Σ	Sigma	Σ	Σ
Τ	Tau	Τ	Τ
Υ	Upsilon	Υ	Υ
Φ	Phi	Φ	Φ
Χ	Chi	Χ	Χ
Ψ	Psi	Ψ	Ψ
Ω	Omega	Ω	Ω
α	alpha	α	α
β	beta	β	β
γ	gamma	γ	γ
δ	delta	δ	δ
ε	epsilon	ε	ε
ζ	zeta	ζ	ζ
η	eta	η	η
θ	theta	θ	θ
ι	iota	ι	ι
κ	kappa	κ	κ
λ	lambda	λ	Λ
μ	mu	μ	μ
ν	nu	ν	Ν
ξ	xi	ξ	ξ
ο	omicron	ο	ο
π	pi	π	π
ρ	rho	ρ	ρ
?	sigmaf	ς	ς
σ	sigma	σ	σ
τ	tau	τ	τ

υ	upsilon	υ	υ
φ	phi	φ	φ
χ	chi	χ	χ
ψ	psi	ψ	ψ
ω	omega	ω	ω
?	theta symbol	ϑ	ϑ
?	upsilon symbol	ϒ	ϒ
?	pi symbol	ϖ	ϖ

HTML 支持的其他实体

结果	描述	实体名称	实体编号
?	capital ligature OE	Œ	Œ
?	small ligature oe	œ	œ
?	capital S with caron	Š	Š
?	small S with caron	š	š
?	capital Y with diaeres	Ÿ	Ÿ
?	f with hook	ƒ	ƒ
?	modifier letter circumflex accent	ˆ	ˆ
?	small tilde	˜	˜
?	en space		
?	em space		
?	thin space		
?	zero width non-joiner		‌
?	zero width joiner		‍
?	left-to-right mark		‎
?	right-to-left mark		‏
–	en dash	–	–
—	em dash	—	—
‘	left single quotation mark	‘	‘
’	right single quotation mark	’	’
?	single low-9 quotation mark	‚	‚

“	left double quotation mark	“	“
”	right double quotation mark	”	”
?	double low-9 quotation mark	„	„
?	dagger	†	†
?	double dagger	‡	‡
?	bullet	•	•
…	horizontal ellipsis	…	…
‰	per mille	‰	‰
′	minutes	′	′
″	seconds	″	″
?	single left angle quotation	‹	‹
?	single right angle quotation	›	›
‾	overline	‾	‾
€	euro	€	€
?	trademark	™	™
←	left arrow	←	←
↑	up arrow	↑	↑
→	right arrow	→	→
↓	down arrow	↓	↓
?	left right arrow	↔	↔
?	carriage return arrow	↵	↵
?	left ceiling	⌈	⌈
?	right ceiling	⌉	⌉
?	left floor	⌊	⌊
?	right floor	⌋	⌋
?	lozenge	◊	◊
?	spade	♠	♠
?	club	♣	♣
?	heart	♥	♥
?	diamond	♦	♦

HTML URL 编码

下面是用 URL 编码形式表示的 ASCII 字符（十六进制格式）。

十六进制格式用于在浏览器和插件中显示非标准的字母和字符。

URL 编码 - 从 %00 到 %8f

ASCII Value	URL-encode	ASCII Value	URL-encode	ASCII Value	URL-encode
?	%00	0	%30	`	%60
	%01	1	%31	a	%61
	%02	2	%32	b	%62
	%03	3	%33	c	%63
	%04	4	%34	d	%64
	%05	5	%35	e	%65
	%06	6	%36	f	%66
	%07	7	%37	g	%67
backspace	%08	8	%38	h	%68
tab	%09	9	%39	i	%69
linefeed	%0a	:	%3a	j	%6a
	%0b	;	%3b	k	%6b
	%0c	<	%3c	l	%6c
c return	%0d	=	%3d	m	%6d
	%0e	>	%3e	n	%6e
	%0f	?	%3f	o	%6f
	%10	@	%40	p	%70
	%11	A	%41	q	%71
	%12	B	%42	r	%72
	%13	C	%43	s	%73
	%14	D	%44	t	%74
	%15	E	%45	u	%75

	%16	F	%46	v	%76
	%17	G	%47	w	%77
	%18	H	%48	x	%78
	%19	I	%49	y	%79
	%1a	J	%4a	z	%7a
	%1b	K	%4b	{	%7b
	%1c	L	%4c	|	%7c
	%1d	M	%4d	}	%7d
	%1e	N	%4e	~	%7e
	%1f	O	%4f		%7f
space	%20	P	%50	€	%80
!	%21	Q	%51		%81
"	%22	R	%52	?	%82
#	%23	S	%53	?	%83
$	%24	T	%54	?	%84
%	%25	U	%55	…	%85
&	%26	V	%56	?	%86
'	%27	W	%57	?	%87
(	%28	X	%58	?	%88
)	%29	Y	%59	‰	%89
*	%2a	Z	%5a	?	%8a
+	%2b	[	%5b	?	%8b
,	%2c	\	%5c	?	%8c
-	%2d	]	%5d		%8d
.	%2e	^	%5e	?	%8e
/	%2f	_	%5f		%8f

URL 编码 - 从 %90 到 %ff

ASCII Value	URL-encode	ASCII Value	URL-encode	ASCII Value	URL-encode
	%90	à	%c0	e	%f0

‘	%91	á	%c1	?	%f1
’	%92	?	%c2	ò	%f2
“	%93	?	%c3	ó	%f3
”	%94	?	%c4	?	%f4
?	%95	?	%c5	?	%f5
–	%96	?	%c6	?	%f6
—	%97	?	%c7	÷	%f7
?	%98	è	%c8	?	%f8
?	%99	é	%c9	ù	%f9
?	%9a	ê	%ca	ú	%fa
?	%9b	?	%cb	?	%fb
?	%9c	ì	%cc	ü	%fc
	%9d	í	%cd	y	%fd
?	%9e	?	%ce	t	%fe
?	%9f	?	%cf	?	%ff
	%a0	D	%d0		
?	%a1	?	%d1		
¢	%a2	ò	%d2		
£	%a3	ó	%d3		
	%a4	?	%d4		
¥	%a5	?	%d5		
|	%a6	?	%d6		
§	%a7		%d7		
¨	%a8	?	%d8		
?	%a9	ù	%d9		
a	%aa	ú	%da		
?	%ab	?	%db		
?	%ac	ü	%dc		
-	%ad	Y	%dd		
?	%ae	T	%de		
-	%af	?	%df		
°	%b0	à	%e0		

±	%b1	á	%e1
2	%b2	a	%e2
3	%b3	?	%e3
′	%b4	?	%e4
µ	%b5	?	%e5
?	%b6	?	%e6
·	%b7	?	%e7
?	%b8	è	%e8
1	%b9	é	%e9
o	%ba	ê	%ea
?	%bb	?	%eb
?	%bc	ì	%ec
?	%bd	í	%ed
?	%be	?	%ee
?	%bf	?	%ef

HTML 语言代码参考手册

ISO 语言代码

HTML 的 lang 属性可用于网页或部分网页的语言。这对搜索引擎和浏览器是有帮助的。

根据 W3C 推荐标准，您应该通过 <html> 标签中的 lang 属性对每张页面中的主要语言进行声明，比如：

```
<html lang="en">
...
</html>
```

在 XHTML 中，采用如下方式在 <html> 标签中对语言进行声明：

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
...
</html>
```

ISO 639-1 语言代码

ISO 639-1 为各种语言定义了缩略词。您可以在 HTML 和 XHTML 中的 lang 和 xml:lang 属性中使用它们。

Language	ISO Code
Abkhazian	ab
Afar	aa
Afrikaans	af
Albanian	sq
Amharic	am
Arabic	ar
Armenian	hy
Assamese	as
Aymara	ay
Azerbaijani	az
Bashkir	ba
Basque	eu

Bengali (Bangla)	bn
Bhutani	dz
Bihari	bh
Bislama	bi
Breton	br
Bulgarian	bg
Burmese	my
Byelorussian (Belarusian)	be
Cambodian	km
Catalan	ca
Cherokee	
Chewa	
Chinese (Simplified)	zh
Chinese (Traditional)	zh
Corsican	co
Croatian	hr
Czech	cs
Danish	da
Divehi	
Dutch	nl
Edo	
English	en
Esperanto	eo
Estonian	et
Faeroese	fo
Farsi	fa
Fiji	fj
Finnish	fi
Flemish	
French	fr
Frisian	fy

Fulfulde	
Galician	gl
Gaelic (Scottish)	gd
Gaelic (Manx)	gv
Georgian	ka
German	de
Greek	el
Greenlandic	kl
Guarani	gn
Gujarati	gu
Hausa	ha
Hawaiian	
Hebrew	he, iw
Hindi	hi
Hungarian	hu
Ibibio	
Icelandic	is
Igbo	
Indonesian	id, in
Interlingua	ia
Interlingue	ie
Inuktitut	iu
Inupiak	ik
Irish	ga
Italian	it
Japanese	ja
Javanese	jv
Kannada	kn
Kanuri	
Kashmiri	ks
Kazakh	kk
Kinyarwanda (Ruanda)	rw

Kirghiz	ky
Kirundi (Rundi)	rn
Konkani	
Korean	ko
Kurdish	ku
Laothian	lo
Latin	la
Latvian (Lettish)	lv
Limburgish (Limburger)	li
Lingala	ln
Lithuanian	lt
Macedonian	mk
Malagasy	mg
Malay	ms
Malayalam	ml
Maltese	mt
Maori	mi
Marathi	mr
Moldavian	mo
Mongolian	mn
Nauru	na
Nepali	ne
Norwegian	no
Occitan	oc
Oriya	or
Oromo (Afan, Galla)	om
Papiamentu	
Pashto (Pushto)	ps
Polish	pl
Portuguese	pt
Punjabi	pa

Quechua	qu
Rhaeto-Romance	rm
Romanian	ro
Russian	ru
Sami (Lappish)	
Samoan	sm
Sangro	sg
Sanskrit	sa
Serbian	sr
Serbo-Croatian	sh
Sesotho	st
Setswana	tn
Shona	sn
Sindhi	sd
Sinhalese	si
Siswati	ss
Slovak	sk
Slovenian	sl
Somali	so
Spanish	es
Sundanese	su
Swahili (Kiswahili)	sw
Swedish	sv
Syriac	
Tagalog	tl
Tajik	tg
Tamazight	
Tamil	ta
Tatar	tt
Telugu	te
Thai	th
Tibetan	bo

Tigrinya	ti
Tonga	to
Tsonga	ts
Turkish	tr
Turkmen	tk
Twi	tw
Uighur	ug
Ukrainian	uk
Urdu	ur
Uzbek	uz
Venda	
Vietnamese	vi
Volapuk	vo
Welsh	cy
Wolof	wo
Xhosa	xh
Yi	
Yiddish	yi, ji
Yoruba	yo
Zulu	zu

HTTP 状态消息

当浏览器从 **web** 服务器请求服务时，可能会发生错误。

从而有可能会返回下面的一系列状态消息：

1xx: 信息

消息:	描述:
100 Continue	服务器仅接收到部分请求，但是一旦服务器并没有拒绝该请求，客户端应该继续发送其余的请求。
101 Switching Protocols	服务器转换协议：服务器将遵从客户的请求转换到另外一种协议。

2xx: 成功

消息:	描述:
200 OK	请求成功（其后是对GET和POST请求的应答文档。）
201 Created	请求被创建完成，同时新的资源被创建。
202 Accepted	供处理的请求已被接受，但是处理未完成。
203 Non-authoritative Information	文档已经正常地返回，但一些应答头可能不正确，因为使用的是文档的拷贝。
204 No Content	没有新文档。浏览器应该继续显示原来的文档。如果用户定期地刷新页面，而Servlet可以确定用户文档足够新，这个状态代码是很有用的。
205 Reset Content	没有新文档。但浏览器应该重置它所显示的内容。用来强制浏览器清除表单输入内容。
206 Partial Content	客户发送了一个带有Range头的GET请求，服务器完成了它。

3xx: 重定向

消息:	描述:
300 Multiple Choices	多重选择。链接列表。用户可以选择某链接到达目的地。最多允许五个地址。
301 Moved Permanently	所请求的页面已经转移至新的url。
302 Found	所请求的页面已经临时转移至新的url。
303 See Other	所请求的页面可在别的url下被找到。
304 Not Modified	未按预期修改文档。客户端有缓冲的文档并发出了一个条件性的请求（一般是提供If-Modified-Since头表示客户只想比指定日期更新的文档）。服务器告诉客户，原来缓冲的文档还可以继续使用。
305 Use Proxy	客户请求的文档应该通过Location头所指明的代理服务器提取。
306 *Unused*	此代码被用于前一版本。目前已不再使用，但是代码依然被保留。
307 Temporary Redirect	被请求的页面已经临时移至新的url。

4xx: 客户端错误

消息:	描述:
400 Bad Request	服务器未能理解请求。
401 Unauthorized	被请求的页面需要用户名和密码。
402 Payment Required	此代码尚无法使用。
403 Forbidden	对被请求页面的访问被禁止。
404 Not Found	服务器无法找到被请求的页面。
405 Method Not Allowed	请求中指定的方法不被允许。
406 Not Acceptable	服务器生成的响应无法被客户端所接受。
407 Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408 Request Timeout	请求超出了服务器的等待时间。
409 Conflict	由于冲突，请求无法被完成。
410 Gone	被请求的页面不可用。
411 Length Required	"Content-Length" 未被定义。如果无此内容，服务器不会接受请求。
412 Precondition Failed	请求中的前提条件被服务器评估为失败。
413 Request Entity Too Large	由于所请求的实体的太大，服务器不会接受请求。
414 Request-url Too Long	由于url太长，服务器不会接受请求。当post请求被转换为带有很长的查询信息的get请求时，就会发生这种情况。
415 Unsupported Media Type	由于媒介类型不被支持，服务器不会接受请求。
416	服务器不能满足客户在请求中指定的Range头。
417 Expectation Failed	

5xx: 服务器错误

消息:	描述:
500 Internal Server Error	请求未完成。服务器遇到不可预知的情况。
501 Not Implemented	请求未完成。服务器不支持所请求的功能。
502 Bad Gateway	请求未完成。服务器从上游服务器收到一个无效的响应。
503 Service Unavailable	请求未完成。服务器临时过载或当机。
504 Gateway Timeout	网关超时。
505 HTTP Version Not Supported	服务器不支持请求中指明的HTTP协议版本。

HTTP 方法：GET 对比 POST

两种最常用的 **HTTP** 方法是：**GET** 和 **POST**。

什么是 HTTP？

超文本传输协议（HTTP）的设计目的是保证客户机与服务器之间的通信。

HTTP 的工作方式是客户机与服务器之间的请求-应答协议。

web 浏览器可能是客户端，而计算机上的网络应用程序也可能作为服务器端。

举例：客户端（浏览器）向服务器提交 HTTP 请求；服务器向客户端返回响应。响应包含关于请求的状态信息以及可能被请求的内容。

两种 HTTP 请求方法：GET 和 POST

在客户机和服务器之间进行请求-响应时，两种最常被用到的方法是：GET 和 POST。

- *GET* - 从指定的资源请求数据。
- *POST* - 向指定的资源提交要被处理的数据

GET 方法

请注意，查询字符串（名称/值对）是在 GET 请求的 URL 中发送的：

```
/test/demo_form.asp?name1=value1&name2=value2
```

有关 GET 请求的其他一些注释：

- GET 请求可被缓存
- GET 请求保留在浏览器历史记录中
- GET 请求可被收藏为书签
- GET 请求不应在处理敏感数据时使用
- GET 请求有长度限制
- GET 请求只应当用于取回数据

POST 方法

请注意，查询字符串（名称/值对）是在 POST 请求的 HTTP 消息主体中发送的：

```
POST /test/demo_form.asp HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2
```

有关 POST 请求的其他一些注释：

- POST 请求不会被缓存
- POST 请求不会保留在浏览器历史记录中
- POST 不能被收藏为书签
- POST 请求对数据长度没有要求

比较 GET 与 POST

下面的表格比较了两种 HTTP 方法：GET 和 POST。

	GET	POST
后退按钮/刷新	无害	数据会被重新提交（浏览器应该告知用户数据会被重新提交）。
书签	可收藏为书签	不可收藏为书签
缓存	能被缓存	不能缓存
编码类型	application/x-www-form-urlencoded	application/x-www-form-urlencoded 或 multipart/form-data。为二进制数据使用多重编码。
历史	参数保留在浏览器历史中。	参数不会保存在浏览器历史中。
对数据长度的限制	是的。当发送数据时，GET 方法向 URL 添加数据；URL 的长度是受限制的（URL 的最大长度是 2048 个字符）。	无限制。
对数据类型的限制	只允许 ASCII 字符。	没有限制。也允许二进制数据。
安全性	与 POST 相比，GET 的安全性较差，因为所发送的数据是 URL 的一部分。在发送密码或其他敏感信息时绝不要使用 GET ！	POST 比 GET 更安全，因为参数不会被保存在浏览器历史或 web 服务器日志中。
可见性	数据在 URL 中对所有人都是可见的。	数据不会显示在 URL 中。

其他 HTTP 请求方法

下面的表格列出了其他一些 HTTP 请求方法：

方法	描述
HEAD	与 GET 相同，但只返回 HTTP 报头，不返回文档主体。
PUT	上传指定的 URI 表示。
DELETE	删除指定资源。
OPTIONS	返回服务器支持的 HTTP 方法。
CONNECT	把请求连接转换到透明的 TCP/IP 通道。

免责声明

W3School提供的内容仅用于培训。我们不保证内容的正确性。通过使用本站内容随之而来的风险与本站无关。W3School简体中文版的所有内容仅供测试，对任何法律问题及风险不承担任何责任。